

# ПРОГРАММНАЯ ИНЖЕНЕРИЯ SOFTWARE ENGINEERING

УДК 004.415.2

**В. А. Стенников**, д-р техн. наук, зам. директора по науке, e-mail: sva@isem.sei.irk.ru,  
**Е. А. Барахтенко**, канд. техн. наук, ст. науч. сотр., e-mail: barakhtenko@isem.sei.irk.ru,  
**Д. В. Соколов**, канд. техн. наук, науч. сотр., e-mail: sokolov\_dv@isem.sei.irk.ru,  
Федеральное государственное бюджетное учреждение науки,  
Институт систем энергетики им. Л. А. Мелентьева,  
Сибирского отделения Российской академии наук (ИСЭМ СО РАН), г. Иркутск

## Методический подход на основе концепции Model-Driven Engineering и онтологий к разработке программного обеспечения для проектирования теплоснабжающих систем

*Предложен методический подход к разработке программного обеспечения, основанный на применении концепции Model-Driven Engineering и онтологий. Этот подход позволяет автоматически получать сложные программные системы для оптимального проектирования теплоснабжающих систем с учетом особенностей решаемых прикладных задач и индивидуальных условий развития этих систем. На его базе разработан программный комплекс СОСНА для оптимизации параметров теплоснабжающих систем.*

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

### Введение

В настоящей статье описан разработанный авторами методический подход к построению программного обеспечения для решения задач проектирования теплоснабжающих систем (ТСС). Этот подход основан на применении концепции Model-Driven Engineering ("управляемая моделями разработка", MDE), которая представляет собой совокупность методических принципов автоматического построения сложных программных систем на основе предварительно разработанных моделей [1, 2]. При построении программного обеспечения применяют современные технологии метапрограммирования [3] и формализованные в виде онтологической системы знания предметной области [4]. В данной статье изложены принципиальные положения разработанной в ходе выполненной работы методики автоматического построения программной системы.

В работе представлены принципы построения инструментальной платформы, которая служит в качестве единой основы при автоматическом построении программного обеспечения для решения задач оптимального проектирования ТСС. Приведено описание программного комплекса (ПК) СОСНА, разработанного на базе предложенного методического подхода.

### Описание проблемы и постановка задач исследования

Теоретической основой для решения задач проектирования и управления функционированием трубопроводных систем различного типа и назначения является разработанная и успешно развиваемая в ИСЭМ СО РАН теория гидравлических цепей (ТГЦ), которая является базой для моделирования, расчета, оценивания и оптимизации трубопроводных и гидравлических систем различного типа [5]. В рамках ТГЦ выделяется проблема оптимального проектирования 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 [6]. Особенность решения проблемы оптимального проектирования ТСС состоит в том, что она предполагает формирование своего алгоритма из различного состава подзадач (оптимизация структуры, оптими-



Рис. 1. Алгоритм решения задачи проектирования ТСС

зация параметров, анализ надежности системы, тепловые и гидравлические расчеты, расчет параметров источников теплоты и др.), индивидуального для множества рассматриваемых ТСС, с учетом их конкретных 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 [6].

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 разраба-

тывать 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 про-

граммные 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 внести изменение в существующий алгоритм или создать новый управляющий модуль. 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, 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 в целях повышения их эффективности и надежности функционирования.

### **Новый методический подход**

В результате проведенных авторами исследований предложен новый методический подход к разработке программного обеспечения на базе концепции 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 автоматически строится 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 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 следующее:

- методику автоматического построения сложных программных систем на основе применения концепции MDE, современных методов метапрограммирования и онтологий;
- принципы построения расширяемой архитектуры программного обеспечения, ориентированного на особенности решаемой прикладной задачи;
- принципы построения инструментальной платформы как единой универсальной основы для разработки программного обеспечения решения комплекса задач проектирования ТСС;
- методические принципы разработки и применения универсальных программных компонентов.

В процессе автоматического построения программной системы используются онтологии, которые позволяют формализованно 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 [7]. Существуют работы, в которых описывается опыт применения онтологий при решении задач энергетики [8, 9].

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 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.

### **Методика автоматического построения программной системы**

Предложенная авторами методика автоматического построения программной системы состоит из следующих этапов.

**Этап 1.** Построение компьютерной модели конкретной ТСС.

На этом этапе инженер создает компьютерную модель конкретной ТСС, отражающую свойства сети: структуру, набор используемого оборудования и его свойства, параметры элементов системы (технические, гидравлические, граничные условия) [5]. Эта модель сохраняется в базе данных для многократного использования.

**Этап 2.** Формализация прикладной задачи.

На этом этапе происходит формализация 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 (рис. 2). Эта модель представляет собой совокупность структур данных, описывающих свойства и структуру создаваемой прикладной системы.

Во время этого процесса 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 определение методики решения прикладной задачи, выделение этапов ее решения, создание списка используемых методов и алгоритмов.

5. Создание списка необходимых для решения прикладной задачи программных компонентов на основе списка методов и алгоритмов и онтологии программного обеспечения.

6. На основе описания методики из онтологии задач и описания программных компонентов из онтологии программного обеспечения построение графа решения прикладной задачи, отражающего взаимосвязь этапов решения этой 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 [10]. В рамках методики используются следующие технологии: 1) автоматическая генерация программного кода; 2) динамическая компиляция программного кода; 3) рефлексивное программирование.

Схема взаимодействия программных компонентов при построении программной системы представлена на рис. 3. Путем генерации программного кода и его компиляции создаются программные реализации моделей элементов тепловой сети. С помощью 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 [11].

Во время этого процесса решаются следующие подзадачи.

1. Преобразование математических и графических моделей элементов ТСС, формализованно описанных в

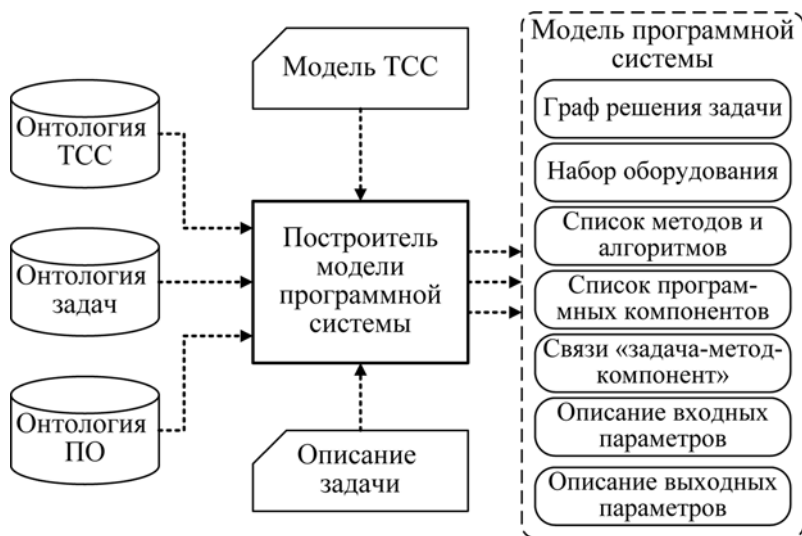


Рис. 2. Схема построения модели программной системы

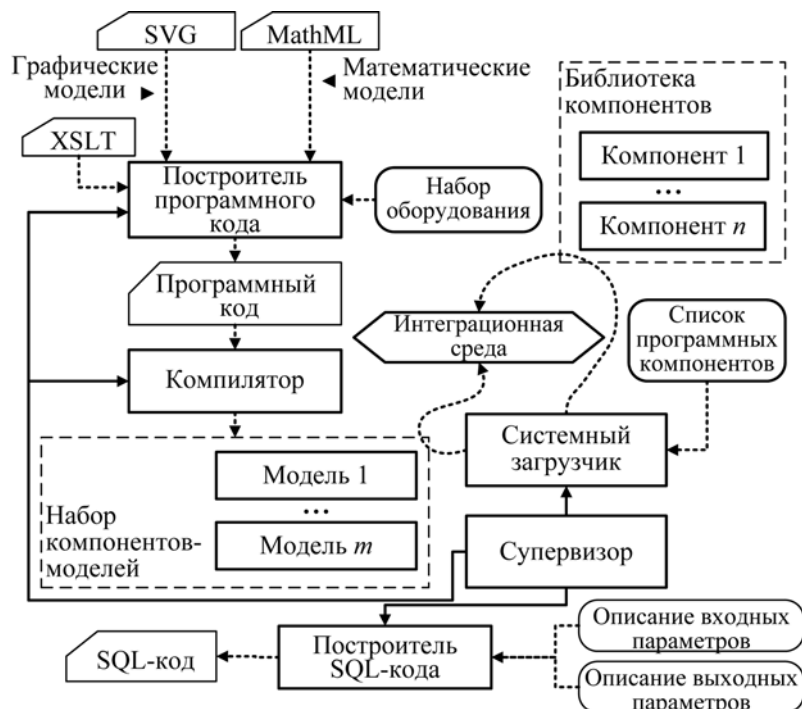


Рис. 3. Схема построения программной системы

форматах MathML<sup>1</sup> и SVG<sup>2</sup>, в программный код на основе правил, формализованных на языке XSLT<sup>3</sup>.

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 [12].

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 рис. 4.

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 с помощью 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.

### Инструментальная платформа

Программная система строится автоматически на основе разработанной инструментальной платформы. В качестве базового языка программирования для реализации инструментальной платформы использован Java. Выбор этого языка обусловлен следующими преимуществами [11—13]: 1) поддержка современных технологий программирования, таких как объектно-ориентированное, компонентное и функциональное программирование; 2) встроенная поддержка технологий метапрограммирования (рефлексивное программирование, возможность использовать компилятор для динамического создания программных компонентов, технология Java Scripting); 3) широкий набор технологий и инструментов для организации распределенных и парал-

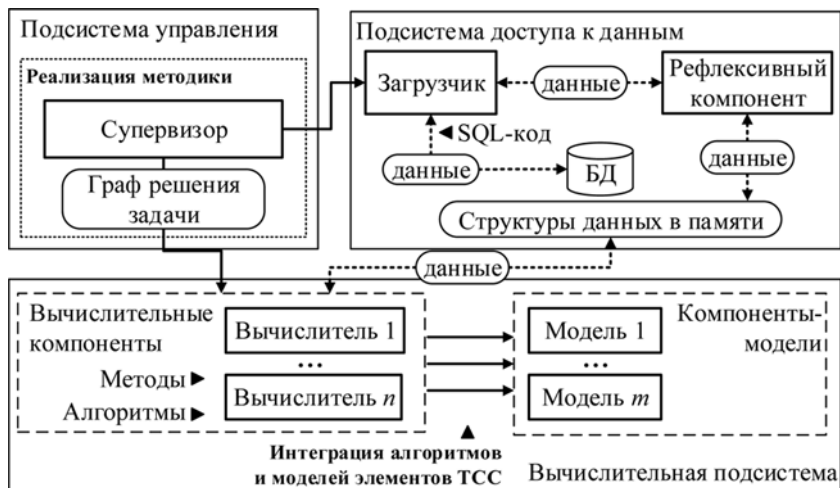


Рис. 4. Архитектура программной системы, получаемой в результате автоматического построения

ельных вычислений; 4) единая универсальная технология доступа к базам данных JDBC. Программные компоненты, реализующие методы и алгоритмы, могут быть реализованы на любых языках программирования (C, C++, Fortran и др.). Интеграция этих компонентов осуществляется через стандартизированные интерфейсы, реализуемые за счет применения паттернов проектирования (англ. *design patterns*) [12]. Применение паттернов Command и Adapter обеспечивает универсальный интерфейс доступа к программным компонентам [10].

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;
- 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

<sup>1</sup> <http://www.w3.org/Math/>

<sup>2</sup> <http://www.w3.org/TR/SVG/>

<sup>3</sup> <http://www.w3.org/TR/xslt/>

решения задач проектирования ТСС, образует библиотеку вычислительных компонентов инструментальной платформы. Компоненты этой группы разрабатывают для решения задачи многократного использования программных реализаций математических методов и алгоритмов в соответствии со следующим принципом: однажды реализованные компоненты можно многократно использовать 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, подготовка и экспертиза 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 схожести итерационного процесса.

Решение прикладных задач проектирования ТСС требует организации сложного итерационного процесса из различных подзадач, поэтому разработанная архитектура включает специальную подсистему — *супервизор*. Эта подсистема отвечает за выполнение следующих функций: 1) вызов программных компонентов, формирующих модель программной системы; 2) вызов программных компонентов, отвечающих за автоматическое построение программной системы; 3) 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 с помощью 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  $S^O$  имеет следующий вид:

$$S^O = (O^M, O^П, \Omega^{MB}),$$

где  $O^M$  — метаонтология;  $O^П$  — множество предметных онтологий;  $\Omega^{MB}$  — модель машины вывода.

Множество предметных онтологий  $O^П$  описывается выражением

$$O^П = \{O^{TCC}, O^3, O^{ПО}\},$$

где  $O^{TCC}$  — онтология теплоснабжающей системы (онтология ТСС);  $O^3$  — онтология задач;  $O^{ПО}$  — 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 рис. 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.

*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

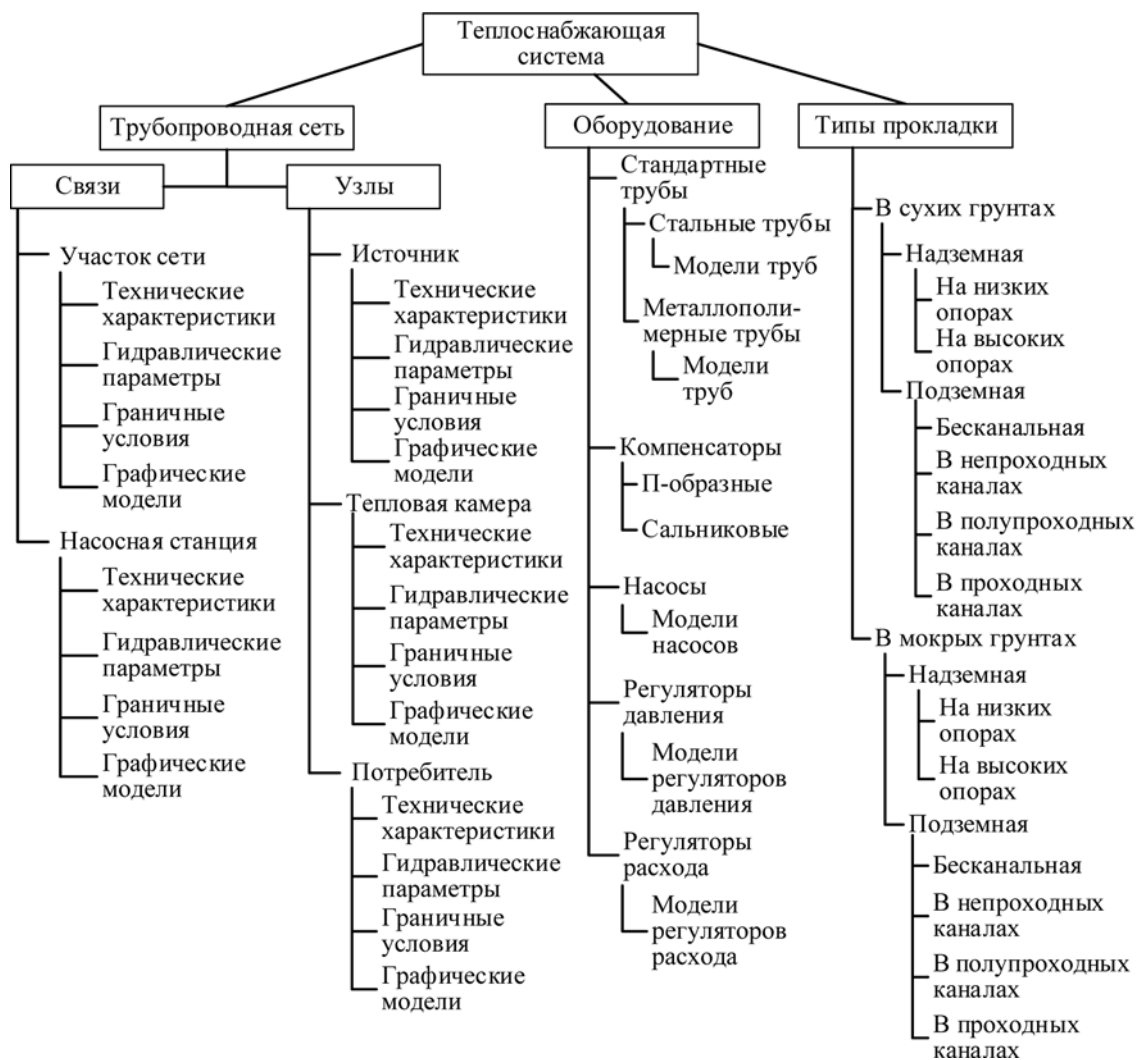


Рис. 5. Фрагмент онтологии ТСС

компоненты-модели, которые используются при проведении математических расчетов и отображении компьютерной модели ТСС во время решения научных и прикладных инженерных задач.

### 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 нового поколения СОСНА (Синтез Оптимальных Систем с учетом надежности) [14], предназначенный для решения сложных инженерных задач оптимизации параметров тепловых сетей, многократно решаемых при проектировании ТСС [5]. ПК СОСНА позволяет определять: 1) "узкие места" в системе и способы их устранения; 2) способы реконструкции существующих участков; 3) диаметры новых участков сети; 4) 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 рис. 4. Применение разработанного методического подхода при реализации ПК СОСНА обеспечило соз-

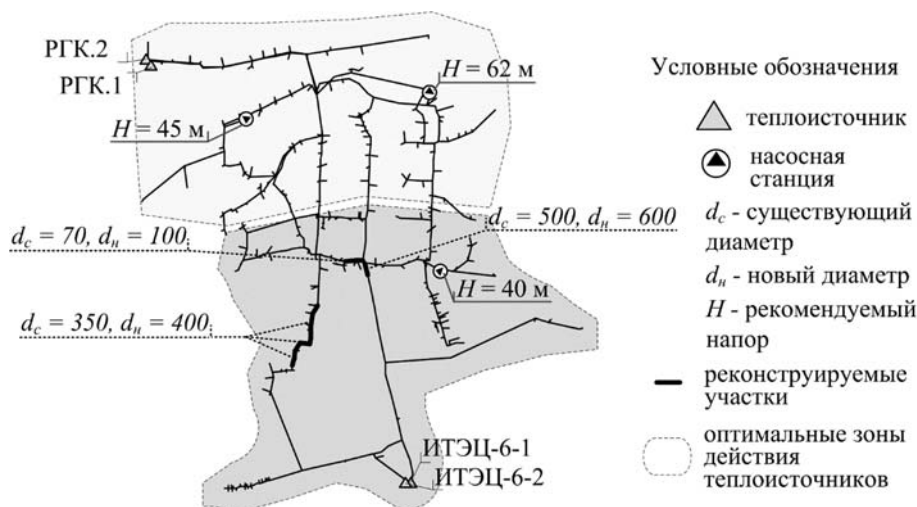


Рис. 6. Схема теплоснабжающей системы г. Братска с обозначенными рекомендованными мероприятиями по ее реконструкции

дание открытой программной системы, приспособленной для развития и сопровождения.

Применение онтологий при реализации программного обеспечения позволило получить следующие возможности:

- формализованно 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 с помощью 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 [5]. Для оптимизации кольцевых сетей реализован метод многоконтурной оптимизации (МКО), основанный на последовательном улучшении решений [5]. 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 были проведены 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, г. Братска и поселка Магистральный. На рис. 6 приведены полученные по результатам расчетов целесообразные мероприятия по реконструкции теплоснабжающей системы г. Братска.

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

## Заключение

Предложен методический подход к разработке программного обеспечения, основанный на применении концепции MDE, который 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. ПК можно применять 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.

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

## Список литературы

1. **Brambilla M., Cabot J., Wimmer M.** Model Driven Software Engineering in Practice: Synthesis Lectures on Software Engineering #1. San Rafael: Morgan & Claypool, 2012. 182 p.
2. **Vöiter M., Stahl T., Bettin J., Haase A., Helsen S.** Model-Driven Software Development: Technology, Engineering, Management. New York: Wiley, 2006. 446 p.
3. **Bartlett J.** The art of metaprogramming, Part I: Introduction to metaprogramming // IBM homepage. URL: <http://www.ibm.com/developerworks/library/l-metaprogl/> (дата обращения: 22.10.2014).
4. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.
5. **Меренков А. П., Хасиев В. Я.** Теория гидравлических цепей. М.: Наука, 1985. 280 с.
6. **Стеников В. А., Сеннова Е. В., Ощепкова Т. Б.** Методы комплексной оптимизации развития теплоснабжающих систем // Изв. РАН. Энергетика. 2006. № 3. С. 44–54.
7. **Евгеньев Г. Б.** Интеллектуальные системы проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 334 с.



8. **Загорулько Ю. А., Загорулько Г. Б.** Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии // Вестник НГУ. Серия "Информационные технологии". Новосибирск: НГУ, 2012. Т. 10. Вып. 1. С. 121—129.

9. **Ворожцова Т. Н., Скрипкин С. К.** Использование онтологий при моделировании программного комплекса // Вычислительные технологии. 2008. Т. 13, ч. 1. С. 376—381.

10. **Стенников В. А., Барактенко Е. А., Соколов Д. В.** Применение метапрограммирования в программном комплексе для решения задач схемно-параметрической оптимизации теплоснабжающих систем // Программная инженерия. 2011. № 6. С. 31—35.

11. **Forman I., Forman N.** Java Reflection in Action. Greenwich: Manning Publications Co., 2005. 273 p.

12. **Martin R. C.** Agile Software Development: Principles, Patterns and Practices. Upper Saddle River: Pearson Education, USA, 2002. 529 p.

13. **Gosling J., Joy B., Steele G., Bracha G., Buckley A.** Java Language Specification, Java SE 8 Edition. Boston: Addison-Wesley Professional, USA, 2014. 792 p.

14. **Стенников В. А., Барактенко Е. А., Соколов Д. В.** Применение онтологий при реализации программного комплекса для решения задач оптимального проектирования теплоснабжающих систем // Информационные технологии. 2013. № 3. С. 2—7.

**V. A. Stennikov**, Professor, Deputy Director, e-mail: sva@isem.sei.irk.ru,  
**E. A. Barakhtenko**, Senior Researcher, e-mail: barakhtenko@isem.sei.irk.ru,  
**D. V. Sokolov**, Researcher, e-mail: sokolov\_dv@isem.sei.irk.ru

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

## A Methodological Approach on the Basis of Model-Driven Engineering and Ontologies for the Development of a Software for Heat Supply System Design

*The authors suggest a methodological approach to the creation of software. The approach is based on the Model-Driven Engineering methodology and ontologies. This approach allow us automatically create complex software for optimal design of heat supply systems taking into account the peculiarity of application problems and the 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.*

**Keywords:** model-Driven Engineering, metaprogramming, ontology, software, component based programming, ontological system, methodological approach, automation of programming, software architecture, software engineering

### References

1. **Brambilla M., Cabot J., Wimmer M.** *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering #1*. San Rafael, Morgan & Claypool Publishers, 2012. 182 p.

2. **Völter M., Stahl T., Bettin J., Haase A., Helsen S.** *Model-Driven Software Development: Technology, Engineering, Management*. New York, Wiley, 2006. 446 p.

3. **Bartlett J.** *The art of metaprogramming, Part 1: Introduction to metaprogramming*. URL: <http://www.ibm.com/developerworks/library/l-metaprogl/> (accessed 22.10.2014).

4. **Gavrilova T. A., Khoroshevskii V. F.** *Bazy znaniy intellektual'nykh sistem* (Knowledge bases of intelligent systems), Saint Petersburg, Piter, 2000, 384 p. (in Russian).

5. **Merenkov A. P., Khasilev V. Ya.** *Teoriya gidravlicheskikh tsepei* (The Theory of Hydraulic Circuits), Moscow, Nauka, 1985, 280 p. (in Russian).

6. **Stennikov V. A., Sennova E. V., Oshchepkova T. B.** *Metody kompleksnoi optimizatsii razvitiya teplosnabzhayushchikh sistem* (Methods of complex optimization for heat supply system development). *Izvestiya RAN: Energetika*. 2006, N. 3. P. 44—54 (in Russian).

7. **Evgenev G. B.** *Intellektual'nye sistemy proektirovaniya* (Intelligent design systems), Moscow, Bauman Moscow State Technical University, 2009. 334 p. (in Russian).

8. **Zagorul'ko Yu. A., Zagorul'ko G. B.** *Ontologicheskii podkhod k razrabotke sistemy, podderzhki prinyatiya reshenii na neftegazodobyvayushchem predpriyatii* (An ontological approach for the develop-

ment of a decision support system for an oil-production enterprise). *Novosibirsk, Vestnik NGU, series "Informatsionnye tekhnologii"*, 2012. Vol. 10, issue 1. P. 121—129 (in Russian).

9. **Vorozhtsova T. N., Skripkin S. K.** *Ispol'zovanie ontologii pri modelirovaniy programmnoy kompleksa* (The use of ontologies for modeling software system), *Novosibirsk, Vychislitel'nye tekhnologii*, 2008. Vol. 13, part I. P. 376—381 (in Russian).

10. **Stennikov V. A., Barakhtenko E. A., Sokolov D. V.** *Primene-nie metaprogrammirovaniya v programmnom komplekse dlya resheniya zadach skhemno-parametricheskoy optimizatsii teplosnabzhayushchikh sistem* (Metaprogramming in the software for solving the problems of heat supply system schematic and parametric optimization). *Programmnaya inzheneriya*. 2011. N. 6. P. 31—35 (in Russian).

11. **Forman I., Forman N.** *Java Reflection in Action*, Greenwich, Manning Publications Co., 2005. 273 p.

12. **Martin R. C.** *Agile Software Development: Principles, Patterns and Practices*. Upper Saddle River, Pearson Education, 2002. 529 p.

13. **Gosling J., Joy B., Steele G., Bracha G., Buckley A.** *Java Language Specification, Java SE 8 Edition*. Boston, Addison-Wesley Professional, 2014. 792 p.

14. **Stennikov V. A., Barakhtenko E. A., Sokolov D. V.** *Primenenie ontologii pri realizatsii programmnoy kompleksa dlya resheniya zadach optimal'nogo proektirovaniya teplosnabzhayushchikh sistem* (Ontologies application for implementation of the software for solving optimal design problems of heat supply systems). *Informatsionnye tekhnologii*. 2013. N. 3. P. 2—7 (in Russian).