

ИНЖИНИРИНГ ОНТОЛОГИЙ

УДК 004.82:004.89

Научная статья

DOI: 10.18287/2223-9537-2023-13-4-548-561



Разработка онтологий для автоматизации вычислительных процессов при проектировании трубопроводных систем энергетики

© 2023, В.А. Стенников, Е.А. Барахтенко ✉, Д.В. Соколов, Г.С. Майоров

Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения РАН (ИСЭМ СО РАН),
Иркутск, Россия

Аннотация

Проектирование трубопроводных систем энергетики (тепло-, водо-, нефте-, газоснабжения и др.) представляет сложную задачу, требующую при её решении применения специализированного программного обеспечения. Предлагается использовать онтологии для структуризации знаний, которые применяются в процессах автоматизации разработки программного обеспечения, организации вычислительного процесса и информационного наполнения пользовательского интерфейса. Предложенный методологический подход включает следующие составляющие: состав онтологий, инструментальные средства реализации онтологий, методики построения онтологий (метаонтологии, онтологии конкретных классов, онтологии проектных задач, онтологии программного обеспечения). Показано, что применение онтологий позволяет получить следующие результаты: единую платформу для исследования и разработки новых методов, алгоритмов, математических моделей трубопроводных систем и их элементов; возможность автоматизации доступа к данным для разных типов систем и решаемых задач; автоматизированное построение программной системы. Представлен пример применения онтологий при разработке программного обеспечения. Приведена архитектура программной системы для решения задач проектирования трубопроводных систем различных типов и назначений.

Ключевые слова: онтологии, энергетика, автоматизация, программное обеспечение, трубопроводные системы, проектирование.

Цитирование: Стенников В.А., Барахтенко Е.А., Соколов Д.В., Майоров Г.С. Разработка онтологий для автоматизации вычислительных процессов при проектировании трубопроводных систем энергетики // Онтология проектирования. 2023. Т.13, №4(50). С.548-561. DOI:10.18287/2223-9537-2023-13-4-548-561.

Финансирование: исследование выполнено в Институте систем энергетики им. Л.А. Мелентьева Сибирского отделения РАН при поддержке Российского научного фонда (грант № 22-29-01611).

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Введение

В настоящее время трубопроводные системы (ТПС) энергетики (тепло-, водо-, нефте-, газоснабжения и др.) представляют собой инженерные сооружения, отличающиеся большими размерами и сложностью. В работах [1, 2] заложены основы теории гидравлических цепей (ТГЦ). Один из основных принципов ТГЦ заключается в том, что разрабатываемые методы и алгоритмы должны допускать применение их для расчёта ТПС различных типов и назначений. Возможность создания таких методов и алгоритмов обусловлена общностью топологических свойств ТПС и законами сохранения массы и энергии. Объектом исследования ТГЦ в основном являются следующие типы ТПС: централизованные теплоснабжающие си-

стемы (ТСС), системы холодоснабжения, системы водоснабжения, системы газоснабжения, системы нефтеснабжения.

Сети, как объект исследования, характеризуются большим количеством трубопроводов, источников, активных элементов (насосов, регуляторов давления и расхода) и разнородных потребителей. Для проектирования сетей необходимо использовать математические модели (ММ), методы и алгоритмы оптимизации, позволяющие находить технологически и экономически обоснованные решения, средства информационной поддержки процессов принятия решений, специализированное программное обеспечение (ПО).

В ИСЭМ СО РАН разрабатываются методические подходы к построению ПО для компьютерного моделирования ТПС. Применение ПО в процессе проектирования требует высокой гибкости его построения, что обусловлено необходимостью моделирования широкого набора оборудования, а также методов и алгоритмов, настраиваемых на конкретную задачу. Знания о различных типах ТПС необходимо сохранить в форме, пригодной для многократного использования при автоматизации построения ПО и управлении вычислительным процессом при решении проектных задач. Для этого необходимо организовать хранение знаний в виде онтологий. В настоящей статье излагается новый подход к построению системы онтологий и приведено описание его составляющих, представлена архитектура программной системы для решения задач проектирования ТПС различных типов и назначений.

1 Постановка задачи

ТПЦ является базой для решения задач проектирования ТПС. В данном исследовании модель ТПС – это иерархически построенный объект, в котором какая-либо подсистема на одном уровне рассматривается как совокупность связанных элементов, а на более высоком уровне – как единый элемент с агрегированными характеристиками. Для решения задач проектирования ТПС применяется специализированное математическое и ПО. Математическое обеспечение включает математические постановки задач проектирования, модели элементов ТПС, методы и алгоритмы. ПО включает программные компоненты, которые реализуют методики проектирования ТПС, модели элементов ТПС, а также решатели – готовые программные разработки, предназначенные для решения конкретных классов задач.

Проектирование осуществляется на основе интеграции представленных на рисунке 1 составляющих в рамках единого процесса моделирования ТПС. Вследствие различия ТПС и их подсистем возникает необходимость автоматизации организации вычислительного процесса для достижения необходимой гибкости в адаптации на особенности конкретной ТПС.

Построение ПО, предназначенного для моделирования ТПС, является завершающим этапом разработки методов, ММ, методик и алгоритмов. Применение этого ПО при решении научно-исследовательских и практических задач приводит к накоплению опыта, который позволяет разрабатывать более точные ММ, уточнять справочную информацию, повышать быстродействие алгоритмов, улучшать сходимость методов, получать оригинальные решения задач. Накопленный опыт фиксируется путём внесения изменений в ПО, что повышает его качество и соответствие реальным инженерным системам. В результате ПО становится единственным средством формализации и хранения накопленного опыта, который недоступен для изучения и использования широкому кругу специалистов.

Описание методических и программных разработок необходимо выполнить в форме, пригодной для многократного использования. Для этого необходимо перейти к парадигмам разработки ПО, которые позволят однажды формализованные знания использовать многократно. Авторами предложено выполнить формальное описание знаний о предметной области

(ПрО) в виде онтологий, что позволит многократно их использовать при построения прикладного ПО.

Сформулирована новая задача разработки подхода к построению системы онтологий, который объединит в онтологическом пространстве описание методического и ПО, используемых для автоматизации построения ПО и решения комплекса задач проектирования ТПС.

Применение онтологий обеспечит получение следующих результатов:

- структуризация задач проектирования ТПС различных типов и назначений, создание единого универсального описания ПрО;
- возможность использования знаний, как активного ресурса при автоматизации построения ПО;
- обеспечение разработки единой платформы для исследования и разработки новых методов, методик, алгоритмов, ММ ТПС и их элементов;
- обеспечение универсальности интерфейсов программной реализации методов, алгоритмов и моделей элементов ТПС, упрощение поддержки этих программных компонентов;
- возможность автоматизации доступа к данным для разных типов ТПС и решаемых задач;
- автоматизированное построение программной системы;
- информационное наполнение пользовательского интерфейса.

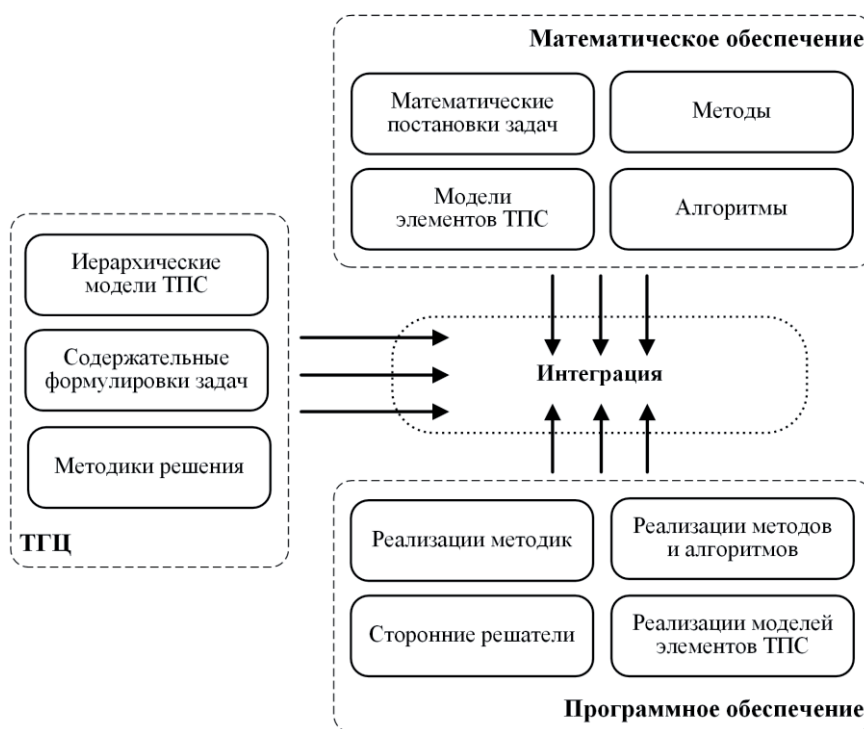


Рисунок 1 – Составляющие моделирования ТПС

2 Обзор литературы

В настоящее время онтологии находят применение в энергетике для описания сложных объектов. В работе [3] рассматривается идея разработки онтологии открытой энергетики для анализа энергетических систем. Исследование в [4] посвящено разработке онтологии, которая содержит описание объектов, относящихся к структурам энергетических сетей и цепочкам спроса и предложения. Предлагаемая онтология описывается с использованием синтаксиса дискрипционной логики, а для её реализации используется язык *OWL2*. Основным результатом работы [5] является разработка онтологии, которая объединяет основные концепции, необходимые для интерпретации всей доступной информации, связанной с рынками электроэнергии. Особенности применения онтологии как основы для формирования информационных моделей в составе цифровых двойников объектов управления интеллектуальной распределённой энергетикой представлены в работе [6]. Онтологическое моделирование при управлении децентрализованными данными в интеллектуальных энергетических системах

рассмотрено в статье [7]. Статья [8] описан фрактальный подход к структурированию знаний, активно используемый для разработки онтологического пространства знаний в области энергетики. В статье [9] представлены онтологии, отражающие основные понятия ситуационного управления, включая ситуационный анализ и ситуационное моделирование, а также вариант онтологии ситуации, рассмотренной с позиции исследования проблемы энергетической безопасности.

В обзоре [10] обсуждаются вопросы проектирования ПО под управлением онтологий. Показано, что в настоящее время направления исследований и разработок смещаются из области технологий программирования в сторону моделирования процессов проектирования ПО.

В программной инженерии активно развиваются концепции, в которых модели ПрО рассматриваются как информационные ресурсы, позволяющие автоматически создавать и развивать программные системы (*Model-Driven Engineering, MDE*) [11]. В статье [12] представлен детальный обзор *MDE*, в котором даны определения модели, метамодели, языка моделирования, программной платформы и программного продукта. Предметно-ориентированные языки (*Domain-Specific Language, DSL*) позволяют создавать детальные описания ПрО, которые можно использовать при автоматизации этапов построения ПО [13, 14]. В методологии программной инженерии (*Ontology Driven Software Engineering, ODSE*) [15, 16] онтология используется на каждом этапе разработки ПО. Использование онтологий на разных этапах отличается по полезности и простоте. В работе [17] показана возможность комплексного использования онтологий. Использование онтологий на всех этапах разработки ПО даёт стратегическое преимущество в расширении обмена и поиске знаний.

В работе [18] предлагается подход к построению ПО для проектирования ТСС на основе концепции *MDE*. Используются технологии метапрограммирования и знания о ПрО в виде онтологий. Приведено описание программного комплекса СОСНА, разработанного на базе изложенного подхода. Программная система строится в автоматизированном режиме с использованием знаний из онтологической системы на основе программных компонентов, реализующих методы и элементы моделируемой ТСС.

3 Подход к построению онтологий

Предложенная схема процесса автоматизированного построения ПО и проведения вычислений при решении задачи проектирования ТПС показана на рисунке 2.

Иерархическая модель ТПС, описание решаемой проектной задачи и программные компоненты, реализующие методы и модели, интегрируются в контексте моделирования конкретной ТПС. Для автоматизации интеграции модели ТПС используется описание класса моделируемой ТПС из соответствующей этому классу онтологии ТПС. Используемое описание позволяет определить: из каких типовых элементов состоит ТПС данного класса, какими параметрами они обладают, из какого оборудования они строятся. Описание задачи из онтологии задач позволяет определить: какую методику необходимо использовать для решения задачи проектирования, какие методы используются в процессе решения. Описание программных компонентов из онтологии ПО позволяет определить: какие программные компоненты будут использоваться, с помощью каких технологий они реализованы, как обеспечить доступ к этим компонентам. Концепция *MDE* позволяет выполнить реализацию предлагаемого решения при автоматизации построения ПО для проектирования ТПС.

Предложенный подход включает в себя следующие составляющие:

- состав онтологий;
- инструментальные средства реализации онтологий;
- методику построения метаонтологии;
- методику построения онтологий ТПС конкретных классов;
- методику построения онтологий задач;
- методику построения онтологий ПО.

В состав онтологий входят следующие: метаонтология, онтологии ТПС конкретных классов, онтологии проектных задач и онтологии ПО.

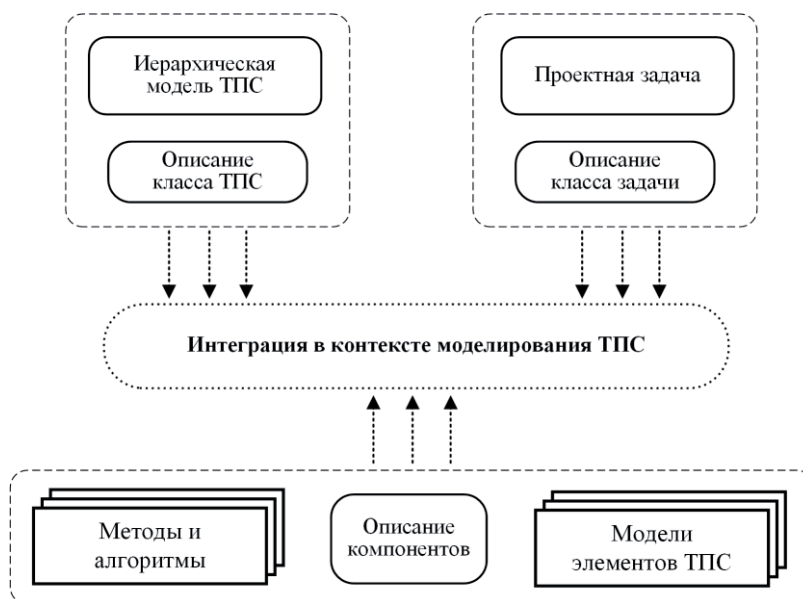


Рисунок 2 – Схема процесса автоматизированного построения ПО

Метаонтология вводит

единое пространство для создания языка описания прикладных онтологий и содержит описание иерархической структуры построения ТПС.

Онтологии ТПС конкретных классов включают следующие составляющие: описание отдельных классов ТПС (тепло-, газо-, водоснабжения и т.д.), классификацию используемого оборудования и его технические характеристики, описание подсистем и элементов ТПС и их свойств (технических характеристик, рабочих параметров и граничных условий), классификацию и описание используемых ММ (например, уравнения, описывающие законы падения напора на ветвях сети, формулы расчёта сопротивлений и др.).

Онтологии задач. Знания о задачах используются для настройки компьютерной модели ТПС на конкретную прикладную задачу и автоматизации процесса подготовки данных для прикладного ПО. Хранение этих знаний организовано в онтологии задач, которая содержит описание прикладных задач и методов их решения, описание алгоритмов, перечисление параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи.

Онтологии ПО. Онтология ПО предназначена для хранения знаний, необходимых для автоматизации построения и использования ПО. Эта онтология содержит описание:

- программных компонентов, реализующих методики, методы и алгоритмы;
- программных компонентов, реализующих модели элементов ТПС;
- программных компонентов, предназначенных для решения конкретных классов задач;
- технологий и интерфейсов доступа к программным компонентам.

Система онтологий реализуется на базе предметно-ориентированного языка, что в соответствии с предлагаемым подходом позволяет автоматизировать процесс программирования при построении ПО, организации вычислительного процесса и информационного наполнения пользовательского интерфейса.

Методика построения метаонтологии включает следующие этапы.

- 1) вводится понятие «Тип данных» и определяются конкретные используемые при моделировании ТПС типы данных.

- 2) определяются базовые для ТПС классы «Трубопроводная система», которая состоит из «Элементов», обладающих «Атрибутами».
- 3) вводятся определения классов «Узлы» и «Ветви», на которые подразделяются «Элементы» ТПС.
- 4) вводится определение класса «Параметр», который в ТГЦ соответствует «Атрибуту».
- 5) вводится определение класса «Иерархическая модель» ТПС, которая отражает «Структурную конфигурацию» и может быть «Древовидной» или «Кольцевой». «Иерархическая модель» описывает набор «Элементов» конкретной моделируемой ТПС.
- 6) вводится определение класса «Проектная задача», обладающего «Входными данными» и «Выходными данными», которые включают в себя «Параметры» элементов моделируемой ТПС.
- 7) вводится определение класса «Модель», реализующего один из «Элементов» ТПС.
- 8) вводится определение класса «Метод».
- 9) вводится определение класса «Методика», которая решает «Проектную задачу» и использует «Метод».
- 10) вводится определение класса «Программный компонент», который может реализовать один из следующих классов: «Методик», «Метод» или «Модель» элемента ТПС.

В соответствии с предложенной методикой вводятся базовые понятия ПрО (см. рисунок 3). Концепт «Трубопроводная система» состоит из «Элементов», обладающих «Атрибутами», которые в ТГЦ называются «Параметрами». Для автоматизации работы с параметрами вводится описание их «Типов данных», включающих «Целые числа», «Числа с плавающей точкой», «Текстовые данные» и др. Решаемая «Проектная задача» имеет «Входные данные» и «Выходные данные», включающие «Параметры» элементов моделируемой системы. Для решения «Проектной задачи» используется «Методика», реализованная в «Программном компоненте».

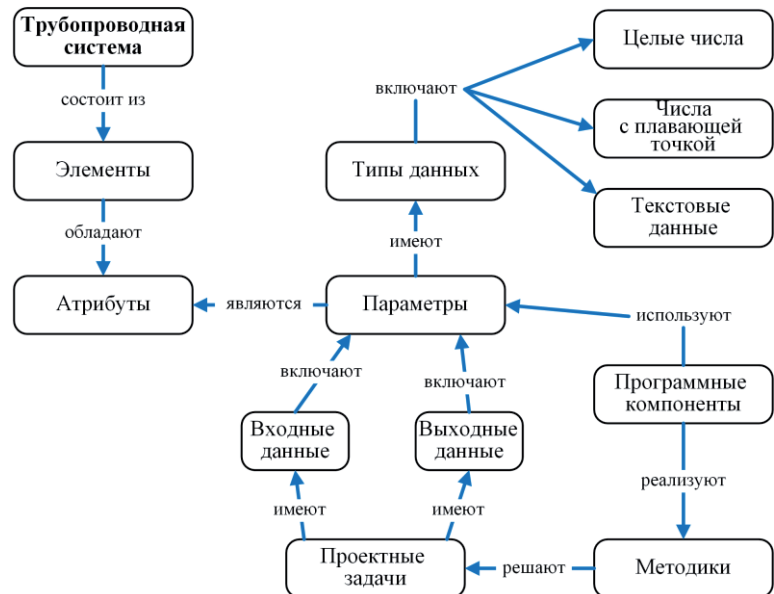


Рисунок 3 – Базовые понятия предметной области

«Трубопроводная система» состоит из «Элементов», которые подразделяются на «Узлы» и «Ветви» (см. рисунок 4). «Трубопроводная система» представляется «Иерархической моделью», отражающей «Структурную конфигурацию», которая может быть «Древовидной» или «Кольцевой». «Программные компоненты» включают программные реализации «Моделей», соответствующих «Элементам», и «Методов», используемых «Методиками». «Проектные задачи» учитывают «Структурную конфигурацию».

Методика построения онтологии ТПС конкретного типа включает следующие этапы.

- 1) вводится определение класса с названием, соответствующим конкретному типу ТПС.
- 2) вводятся определения классов, соответствующих составу «Узлов» для описываемого типа ТПС.

- 3) вводятся определения классов, соответствующих составу «Ветвей» для описываемого типа ТПС.
- 4) для каждого из классов, определённых на этапах 2 и 3, определяются его «Параметры», для которых указываются «Название», «Тип данных», «Ограничения».
- 5) для каждого из классов, определённых на этапах 2 и 3, определяется «Перечень оборудования», используемого при построении определяемого типа ТПС, и описываются его «Характеристики».

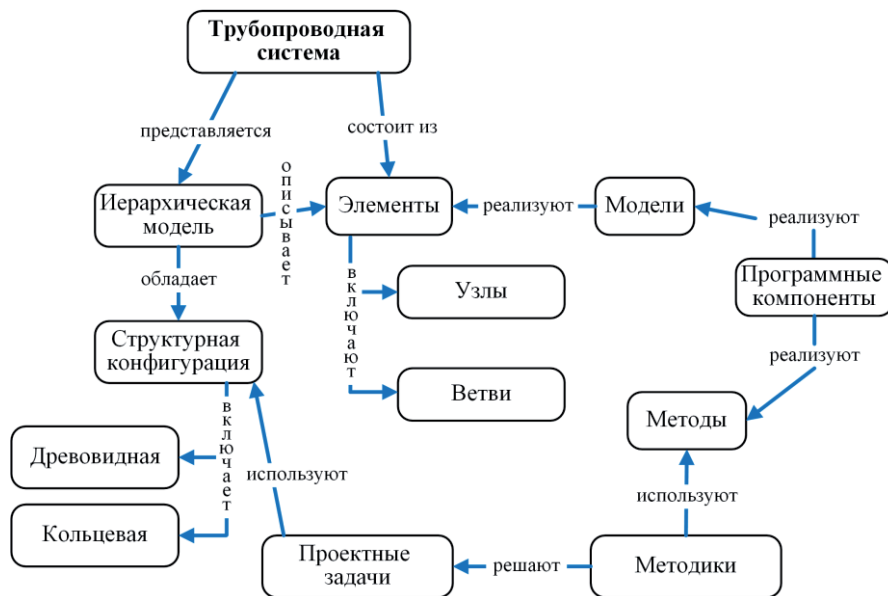


Рисунок 4 – Фрагмент метаонтологии теплоснабжающей системы

Методику построения онтологий ТПС конкретного типа можно показать на примере онтологии ТСС (см. рисунок 5). Эта онтология предназначена для описания построения иерархической модели ТСС и позволяет получить информацию о том, какие типы элементов образуют «Теплоснабжающую систему», какими «Параметрами» обладают «Элементы» и какие «Типы данных» соответствуют «Параметрам». Например, таким элементом является «Источник», обладающий «Напором», «Расходом» и «Геодезической высотой»; «Трубопроводный участок» является «Ветвью» и обладает такими параметрами, как «Потеря напора», «Расход» и «Диаметр».

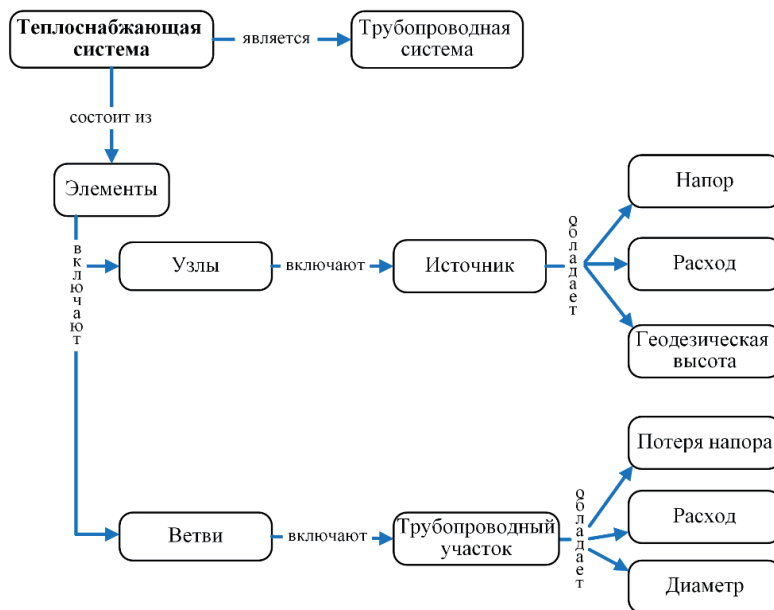


Рисунок 5 – Фрагмент онтологии теплоснабжающей системы

Методика построения онтологии проектных задач включает следующие этапы.

- 1) вводится определение класса задачи с соответствующим ей названием.
- 2) вводится описание «Входных данных» задачи с указанием «Параметров» и их принадлежности конкретным «Элементом» ТПС.

- 3) вводится описание «Выходных данных» задачи с указанием «Параметров» и их принадлежности конкретным «Элементом» ТПС.
- 4) вводятся определения классов, описывающих «Методики» решения определяемой задачи.
- 5) вводятся описания созданных на предыдущем этапе классов, описываются «Методы», используемые каждой из «Методик», и устанавливаются их «Условия применения».
- 6) выполнение этапов 1–5 повторяется, пока не будут определены все классы решаемых задач.

На рисунке 6 приведён фрагмент онтологии задач, который содержит описание задачи оптимизации диаметров трубопроводов ТПС. Вводится концепт задачи «Оптимизация диаметров». «Входными данными» для её решения является «Модель ТПС», а «Выходными данными» – «Диаметры трубопроводов». «Модель ТПС» отражает «Структурную конфигурацию», которая определяет «Метод решения» задачи.

Методика построения онтологии ПО включает следующие этапы.

- 1) вводятся определения классов, соответствующих «Программным компонентам», реализующим «Методики» решения «Проектных задач».
- 2) для каждого из определённых на предыдущем этапе классов описываются необходимые для использования соответствующего компонента характеристики: «Версия», «Технология доступа» и «Язык реализации».
- 3) вводятся определения классов, соответствующих «Программным компонентам», реализующим «Методы» решения «Проектных задач».
- 4) для каждого из определённых на предыдущем этапе классов описываются необходимые для использования соответствующего компонента характеристики: «Версия», «Технология доступа» и «Язык реализации».
- 5) реализуется иерархическое описание классов «Модели» ТПС. Для этого исходное множество реализаций «Моделей» разбивается на подмножества в соответствии с типом ТПС.
- 6) для каждого из полученных на предыдущем этапе подмножеств выполняется определение соответствующих им классов «Программных компонентов», для каждого из которых задаются «Формула», «Технология доступа» и «Язык реализации».

На рисунке 7 представлен фрагмент онтологии ПО, содержащий описание программных компонентов. «Программные компоненты» включают «Реализации моделей ТПС» и «Реализации методов». «Реализации моделей ТПС» включают реализацию моделей «Стальных трубопроводов». Модель «Стального трубопровода» обладает «Формулой», «Технологией доступа» и «Языком реализации». «Реализации методов» включают реализацию «Линейного

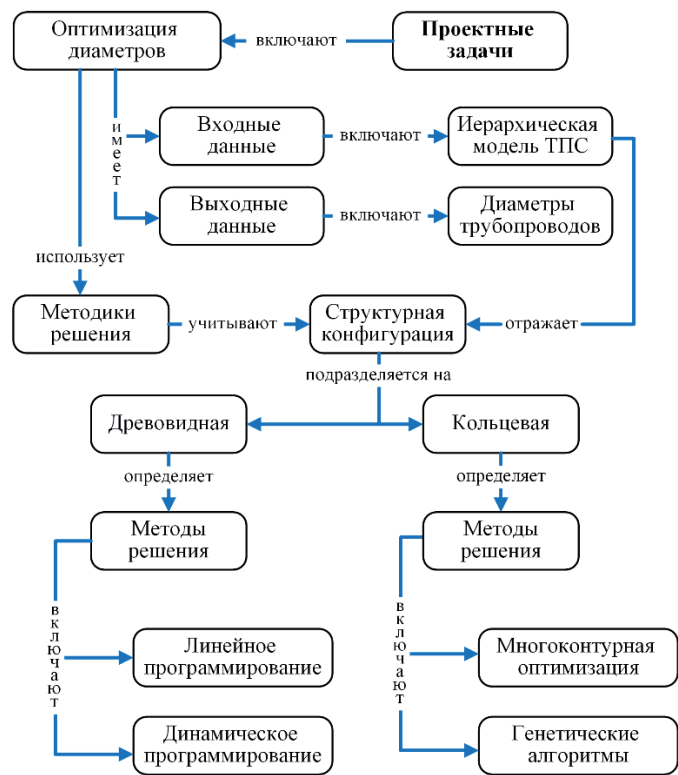


Рисунок 6 – Фрагмент онтологии задач

программирования» в виде компонента «*LpSolve*». Этот компонент обладает «Версией», «Технологией доступа» и «Языком реализации».

4 Практическое применение

Основными составляющими разрабатываемой программной платформы являются следующие: компоненты, управляющие работой других программных компонентов; компоненты, реализующие математические методы и алгоритмы; компоненты для доступа к базам данных; онтологии (метаонтология, онтологии конкретных типов ТПС, онтологии ПО, онтологии задач). *Java* является основным языком программирования для реализации платформы. Отдельные программные компоненты могут быть реализованы на языках *C/C++*, *Fortran*, *Python* и др. Для хранения онтологий используется предметно-ориентированный язык, разработанный на основе языка описания данных *XML*.

Предложенная для реализации программной платформы архитектура многопользовательской сетевой программной системы представлена на рисунке 8.

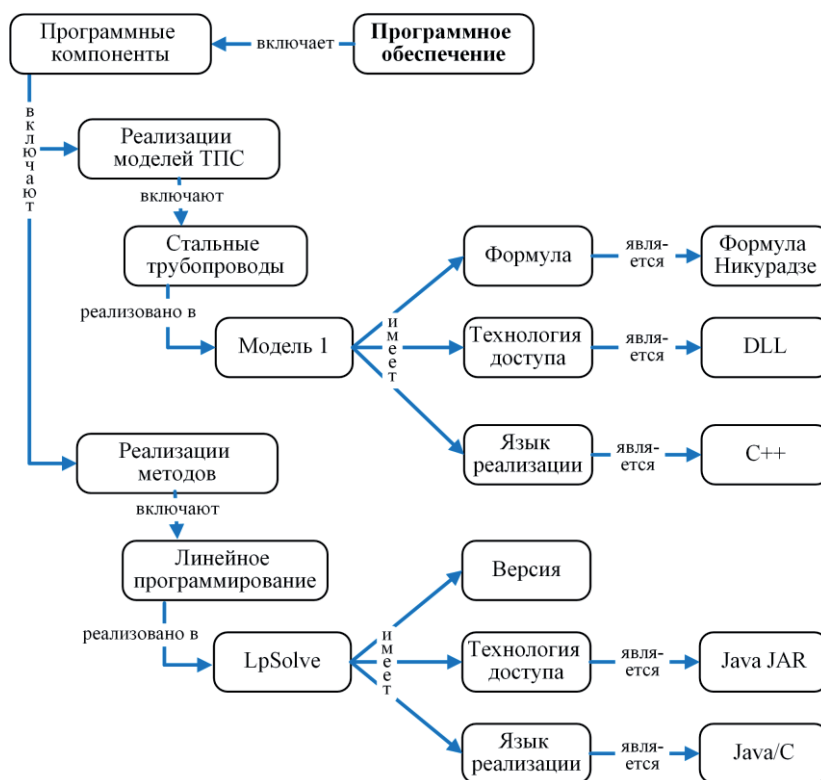


Рисунок 7 – Фрагмент онтологии программного обеспечения

Основные подсистемы разрабатываемой программной системы.

- *Графическая подсистема* обеспечивает создание компьютерной модели ТПС, отражающей структурную конфигурацию системы, исходные характеристики элементов, технические ограничения и проектные рекомендации по построению ТПС. Эта подсистема позволяет пользователю просматривать данные в удобном для восприятия виде и вносить необходимые изменения.
- *Подсистема хранения данных* обеспечивает организацию работы с различными базами данных, применяемыми для хранения и многократного использования компьютерных моделей ТПС, исходных данных и результатов расчётов.

- **Вычислительная подсистема**, которая строится на базе вычислительного ядра в автоматизированном режиме, обеспечивает решение прикладной задачи проектирования ТПС.

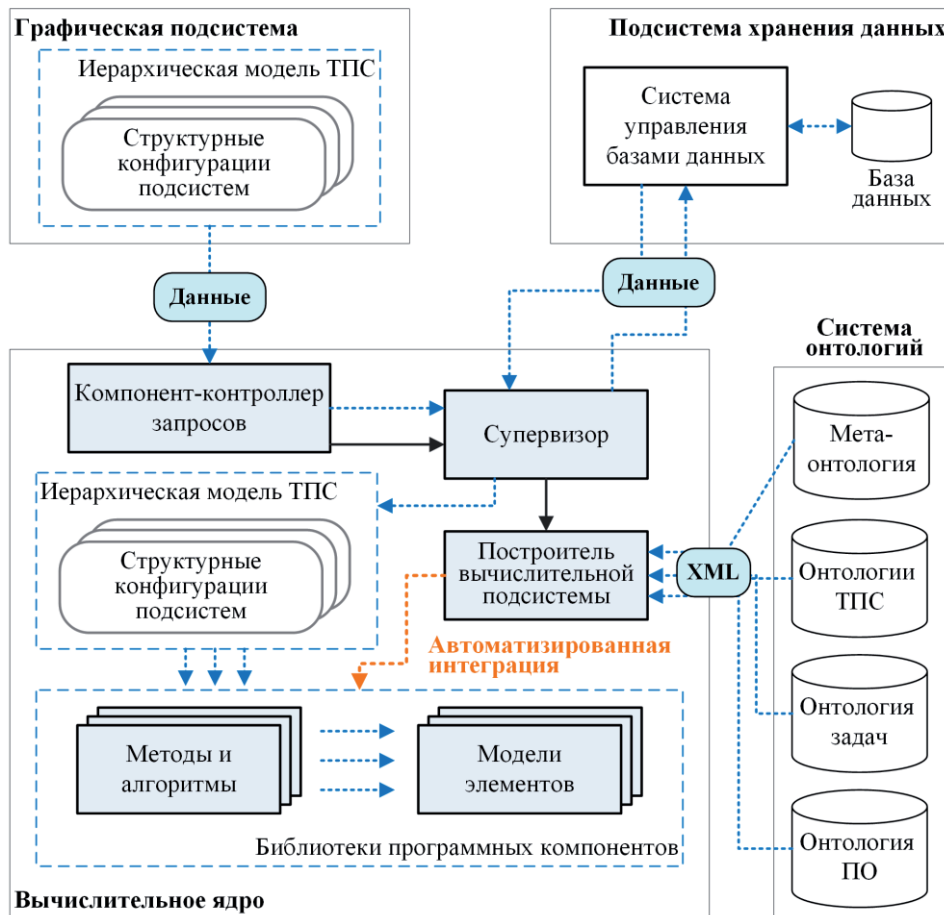


Рисунок 8 – Архитектура многопользовательской сетевой программной системы

Использование онтологий при решении конкретной проектной задачи рассмотрено на примере оптимизации диаметров трубопроводов ТСС. На основе онтологии ТСС автоматически определяется: из каких элементов состоит моделируемая система (см. рисунок 5); какие данные по её параметрам необходимо загрузить из базы данных. После загрузки данных автоматически строится иерархическая модель ТСС. По описанию решаемой задачи из онтологии задач определяется необходимая для использования методика. По описанию методики устанавливается связь между структурной конфигурацией ТСС и необходимым для решения задачи методом (см. рисунок 6). Далее загружается описание необходимых программных компонентов из онтологии ПО (см. рисунок 7), на основе которого определяются необходимые программные компоненты (реализации методики, методов и моделей элементов) и автоматически интегрируются в программную систему. После построения вычислительной подсистемы организуется вычислительный процесс решения прикладной задачи (см. рисунок 1).

Заключение

Построение и практическое применение онтологий позволяет автоматизировать разработку ПО и организацию вычислительных процессов, информационное наполнение пользо-

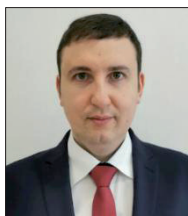
вательского интерфейса, обеспечить эффективную работу с иерархической моделью ТПС при решении задач проектирования ТПС. Предложен подход к построению онтологий для описания знаний о ТПС энергетики различных типов и назначений. Представлен пример применения онтологий при разработке ПО. Приведена архитектура программной системы, которая позволяет выполнять компьютерное моделирование ТПС энергетики.

СПИСОК ИСТОЧНИКОВ

- [1] *Хасилев В.Я.* Элементы теории гидравлических цепей // Изв. АН СССР. Энергетика и транспорт. 1964. №1. С.69-88.
- [2] *Меренков А.П., Хасилев В.Я.* Теория гидравлических цепей. М.: Наука, 1985. 280 с.
- [3] *Booshehri M., Emele L., Flugel S., Forster H., Frey J., Frey U. et al.* Introducing the Open Energy Ontology: Enhancing data interpretation and interfacing in energy systems analysis // Energy and AI, 2021. Vol. 5. 100074. DOI: 10.1016/j.egyai.2021.100074.
- [4] *Devanand A., Karmakar G., Krdzavac N., Rigo-Mariani R., Eddy Y.S., Karimi, I.A. Kraft M.* OntoPowSys: A Power System Ontology for Cross Domain Interactions in an Eco Industrial Park // Energy and AI, 2020. Vol. 1. 100008. DOI: 10.1016/j.egyai.2020.100008.
- [5] *Santos G., Pinto T., Vale Z., Praca I., Morais H.* Electricity markets ontology to support MASCEM's simulations // International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer, 2016. P.393–404.
- [6] *Непша Ф.С., Андриевский А.А., Красильников М.И.* Онтология как основа для создания цифровых двойников объектов управления интеллектуальной распределённой энергетики // Автоматизация в промышленности. 2021. № 1. С.27–33. DOI: 10.25728/avtprom.2021.01.04.
- [7] *Wu J., Orlandi F., AlSkaif T., O'Sullivan D., Dev S.* Ontology Modeling for Decentralized Household Energy Systems // 2021 International Conference on Smart Energy Systems and Technologies (SEST), Vaasa, Finland, 2021. P.1–6. DOI: 10.1109/SEST50973.2021.9543327.
- [8] *Массель Л.В.* Фрактальный подход к структурированию знаний и примеры его применения // Онтология проектирования. 2016. Т.6, №2(20). С.149–161. DOI: 10.18287/2223-9537-2016-6-2-149-161.
- [9] *Массель Л.В., Ворожцова Т.Н., Пяткова Н.И.* Онтологический инжиниринг для поддержки принятия стратегических решений в энергетике // Онтология проектирования. 2017. Т.7, №1(23). С.66–76. DOI: 10.18287/2223-9537-2017-7-1-66-76.
- [10] *Хорошевский В.Ф.* Проектирование систем программного обеспечения под управлением онтологий: модели, методы, реализации // Онтология проектирования. 2019. Т.9, №4(34). С.429–448. DOI: 10.18287/2223-9537-2019-9-4-429-448.
- [11] *Parreiras F.S., Gröner G., Walter T., Friesen A., Rahmani T., Lemcke J., Schwarz H., Miksa K., Wende C., Afmann U.* Model-Driven Software Development. Ontology-Driven Software Development. Springer, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-31226-7_2.
- [12] *Silva da A.R.* Model-driven engineering: A survey supported by the unified conceptual model // Comput. Lang. Syst. Struct., 2015. Vol. 43. P. 139–155. DOI: 10.1016/j.cl.2015.06.001.
- [13] *Fowler M., White T.* Domain-Specific Languages. Addison-Wesley Professional, Denver, 2010.
- [14] *Voelter M.* DSL Engineering: Designing, Implementing and using Domain-Specific Languages. Dslbook, 2013.
- [15] Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. <https://www.w3.org/2001/sw/BestPractices/SE/ODA/060103/>.
- [16] *Pan J.Z., Staab S., Afmann U., Ebert J., Zhao Y.* Ontology-Driven Software Development, 2013. DOI: 10.1007/978-3-642-31226-7.
- [17] *Gonzalez-Perez C.* How Ontologies Can Help in Software Engineering. Lecture Notes in Computer Science, 2017. Vol.10223. DOI: 10.1007/978-3-319-60074-1_2.
- [18] *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. 2014. № 4(14). С.54–68.

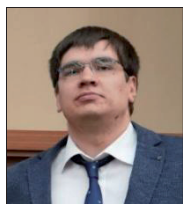
Сведения об авторах

Стенников Валерий Алексеевич, 1954 г. рождения, доктор технических наук (2002 г.), профессор. Академик РАН (2022 г.). Окончил Иркутский политехнический институт. Специалист в области теоретических основ, моделирования, расчёта, оптимизации, управления развитием и функционированием систем энергетики, автор более 250 научных работ, из них 14 монографий. Заслуженный деятель науки Российской Федерации (2011 г.). Лауреат Премии Правительства РФ в области науки и техники (2016 г.). AuthorID (РИНЦ): 442880, Author ID (Scopus): 6508326526, ORCID: 0000-0001-6219-0354. sva@isem.irk.ru.



Барахтенко Евгений Алексеевич, 1982 г. рождения. Окончил Иркутский государственный технический университет в 2005 г., к.т.н. (2011). Старший научный сотрудник ИСЭМ СО РАН. Лауреат Премии Правительства РФ в области науки и техники для молодых ученых (2016 г.). Область научных интересов: математические модели и методы управления развитием трубопроводных систем энергетики и интегрированных энергетических систем, цифровое проектирование. Автор и соавтор более 150 научных работ. Author ID (РИНЦ): 670523; Author ID (Scopus): 55229371000; Researcher ID (WoS): Q-7173-2016. barakhtenko@isem.irk.ru.

Соколов Дмитрий Витальевич, 1984 г. рождения. Окончил Иркутский государственный аграрный университет имени А.А. Ежевского в 2008 г. по спец. «Прикладная информатика (в экономике)», к.т.н. (2013), с.н.с. ИСЭМ СО РАН. Лауреат Премии Правительства РФ в области науки и техники для молодых ученых (2016 г.). В списке научных трудов более 100 работ в области разработки алгоритмов оптимизации трубопроводных систем и автоматизации программирования. Author ID (РИНЦ): 710020; Author ID (Scopus): 57190607698; Researcher ID (WoS): K-6737-2018. sokolov_dv@isem.irk.ru.



Майоров Глеб Сергеевич, 1994 г. рождения. Окончил Иркутский национальный исследовательский технический университет в 2018 г., м.н.с. ИСЭМ СО РАН. Область научных интересов: интегрированные энергетические системы и методы для решения задач их проектирования, мультиагентный подход и его применение. В списке научных трудов более 20 работ. Author ID (РИНЦ): 1001072; Author ID (Scopus): 57211071672; ORCID: 0000-0002-7405-1965. mayorovgs@isem.irk.ru.

Поступила в редакцию 13.11.2023, после рецензирования 24.11.2023. Принята к публикации 1.12.2023.



Scientific article

DOI: 10.18287/2223-9537-2023-13-4-548-561

Development of ontologies for automating computational processes in the energy pipeline system design

© 2023, V.A. Stennikov, E.A. Barakhtenko✉, D.V. Sokolov, G.S. Mayorov

Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia

Abstract

Designing pipeline energy systems (heat, water, oil, gas supply, etc.) is a complex task that requires the use of specialized software to solve it. It is proposed to use ontologies to structure knowledge that is used in the processes of software development automation and organization of the computing process and user interface content. The proposed methodological approach includes the following components: the composition of ontologies, tools for implementing ontologies, and the methods for constructing ontologies (metaontologies, ontologies of specific classes, ontologies of design tasks, software ontologies). It is shown that the use of ontologies allows one to obtain the following results: a single platform

for research and development of new methods, algorithms, mathematical models of pipeline systems and their elements; the ability to automate access to data for different types of systems and tasks; and automated software system construction. An example of ontology system application in software development is presented. The architecture of a software system for solving problems of designing pipeline systems of various types and purposes is given.

Keywords: ontologies, energy, automation, software, pipeline systems, designing.

For citation: Stennikov VA, Barakhtenko EA, Sokolov DV, Mayorov GS. Development of ontologies for automating computational processes in the energy pipeline system design [In Russian]. *Ontology of designing*. 2023; 13(4): 548-561. DOI: 10.18287/2223-9537-2023-13-4-548-561.

Financial Support: The research was performed at the Melentiev Energy Systems Institute of Siberia Branch of the Russian Academy of Sciences under the support of Russian Science Foundation (Grant number 22-29-01611).

Conflict of interest: The authors declare no conflict of interest.

List of figures

- Figure 1 - Components of the methodology for the energy pipeline system design
- Figure 2 - Illustration of the automated software development
- Figure 3 - Basic concepts of the subject area
- Figure 4 - Fragment of the metaontology of the heat supply system
- Figure 5 - Fragment of the ontology of the heat supply system
- Figure 6 - Fragment of the task ontology
- Figure 7 - Fragment of the software ontology
- Figure 8 - Architecture of a multi-user network software system

References

- [1] **Khasilev VYa.** Elements of the theory of hydraulic circuits [In Russian]. *Izvestiya AN SSSR. Energetika i transport*. 1964; 1: 69-88.
- [2] **Merenkov AP, Khasilev VYa.** Theory of hydraulic circuits [In Russian]. *Nauka*, 1985. 280 p.
- [3] **Booshehri M, Emele L, Flugel S, Forster H, Frey J, Frey U. et al.** Introducing the Open Energy Ontology: Enhancing data interpretation and interfacing in energy systems analysis. *Energy and AI*. 2021; 5: 100074. DOI: 10.1016/j.egyai.2021.100074.
- [4] **Devanand A, Karmakar G, Krdzavac N, Rigo-Mariani R, Eddy YS, Karimi IA, Kraft M.** OntoPowSys: A Power System Ontology for Cross Domain Interactions in an Eco Industrial Park. *Energy and AI*. 2020; 1: 100008. DOI: 10.1016/j.egyai.2020.100008.
- [5] **Santos G, Pinto T, Vale Z, Praca I, Morais H.** Electricity markets ontology to support MASCEM's simulations. *International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer*. 2016. P.393–404.
- [6] **Nepsha FS, Andrievsky AA, Krasilnikov MI.** Ontology as a development framework for digital twins of control plants in intelligent distributed power generation [In Russian]. *Automation in industry*. 2021; 1: 27–33. DOI: 10.25728/avtprom.2021.01.04.
- [7] **Wu J, Orlandi F, AlSkaif T, O'Sullivan D, Dev S.** Ontology Modeling for Decentralized Household Energy Systems. *2021 International Conference on Smart Energy Systems and Technologies (SEST), Vaasa, Finland*. 2021: 1–6. DOI: 10.1109/SEST50973.2021.9543327.
- [8] **Massel LV.** Fractal approach to knowledge structuring and examples of its application [In Russian]. *Ontology of designing*. 2016; 6(2): 149–161. DOI: 10.18287/2223-9537-2016-6-2-149-161.
- [9] **Massel LV, Vorozhtsova TN, Pjatkova NI.** Ontology engineering to support strategic decision-making in the energy sector [In Russian]. *Ontology of designing*. 2017; 7(1): 66–76. DOI: 10.18287/2223-9537-2017-7-1-66-76.
- [10] **Khoroshevsky VF.** Ontology Driven Software Engineering: Models, Methods, Implementations [In Russian]. *Ontology of designing*. 2019; 9(4): 429–448. DOI: 10.18287/2223-9537-2019-9-4-429-448.
- [11] **Parreiras FS, Gröner G, Walter T, Friesen A, Rahmani T, Lemcke J, Schwarz H, Miksa K, Wende C, Aßmann U.** Model-Driven Software Development. *Ontology-Driven Software Development*. Springer, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-31226-7_2.
- [12] **Silva da AR.** Model-driven engineering: A survey supported by the unified conceptual model. *Comput. Lang. Syst. Struct.* 2015; 43: 139–155. DOI: 10.1016/j.cl.2015.06.001.
- [13] **Fowler M, White T.** Domain-Specific Languages. Addison-Wesley Professional, Denver, 2010.

- [14] **Voelter M.** DSL Engineering: Designing, Implementing and using Domain-Specific Languages. Dslbook, 2013.
- [15] Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. <https://www.w3.org/2001/sw/BestPractices/SE/ODA/060103/>.
- [16] **Pan JZ, Staab S, Afmann U, Ebert J, Zhao Y.** Ontology-Driven Software Development, 2013. DOI: 10.1007/978-3-642-31226-7.
- [17] **Gonzalez-Perez C.** How Ontologies Can Help in Software Engineering. Lecture Notes in Computer Science. 2017; 10223. DOI: 10.1007/978-3-319-60074-1_2.
- [18] **Stennikov VA, Barakhtenko EA, Sokolov DV.** Usage of ontologies in the implementation of the concept of model-driven engineering for the design of heat supply systems [In Russian]. *Ontology of designing*. 2014; 4(14): 54–68.
-

About the authors

Valery Alekseevich Stennikov (b. 1954), Doctor of Technical Sciences (2002), Professor. Academician of the Russian Academy of Sciences (2022). Graduated from the Irkutsk Polytechnic Institute. He is a specialist in the field of theoretical foundations, modelling, calculation, optimization, management of development and functioning of energy systems, author of more than 250 scientific papers, including 14 monographs. Honoured Scientist of the Russian Federation (2011). Laureate of the Prize of the Government of the Russian Federation in the field of science and technology (2016). AuthorID (RSCI): 442880, Author ID (Scopus): 6508326526, ORCID: 0000-0001-6219-0354. sva@isem.irk.ru.

Evgeny Alekseevich Barakhtenko (b. 1982) graduated from the Irkutsk State Technical University in 2005, PhD (2011). He is a Senior Researcher at the Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences. Laureate of the Prize of the Government of the Russian Federation in the field of science and technology for young scientists (2016). His scientific interests include mathematical models and methods for expansion planning of energy pipeline systems and integrated energy systems, and digital design. Evgeny Barakhtenko is the author and a coauthor of more than 150 scientific papers. Author ID (RSCI): 670523; Author ID (Scopus): 55229371000; Researcher ID (WoS): Q-7173-2016. barakhtenko@isem.irk.ru. ✉.

Dmitry Vitalyevich Sokolov (b. 1984) graduated from the Irkutsk State Agrarian University named after A.A. Ezhevsky (Irkutsk, Russia) in 2008, PhD (2013). Laureate of the Prize of the Government of the Russian Federation in the field of science and technology for young scientists (2016). The list of scientific works includes more than 100 works in the field of development of algorithms for optimisation of pipeline systems and automation of programming. Author ID (RSCI): 710020; Author ID (Scopus): 57190607698; Researcher ID (WoS): K-6737-2018. sokolov_dv@isem.irk.ru.

Gleb Sergeevich Mayorov (b. 1994) graduated from the Irkutsk National Research Technical University (Irkutsk, Russia) in 2018, a junior researcher in MESI SB RAS. The research interests are integrated energy systems and methods for solving their design problems, multi-agent approach and its application. The list of scientific works includes more than 20 works. Author ID (RSCI): 1001072; Author ID (Scopus): 57211071672; ORCID: 0000-0002-7405-1965. mayorovgs@isem.irk.ru.

Received November 13, 2023. Revised November 24, 2023. Accepted December 1, 2023.
