

В. А. Стенников, чл.-корр., директор, e-mail: sva@isem.irk.ru,
Е. А. Барахтенко, канд. техн. наук, ст. науч. сотр., e-mail: barakhtenko@isem.irk.ru,
Д. В. Соколов, канд. техн. наук, ст. науч. сотр., e-mail: sokolov_dv@isem.irk.ru,
Федеральное государственное бюджетное учреждение науки
Институт систем энергетики им. Л. А. Мелентьева
Сибирского отделения Российской академии наук (ИСЭМ СО РАН), г. Иркутск

Разработка принципов построения интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики*

Представлен методический подход к построению интегрированной графической среды для компьютерного моделирования трубопроводных систем энергетики различного типа. Автоматизированное построение интегрированной графической среды выполняется на основе компьютерной модели трубопроводной системы и онтологий на базе концепции Model-Driven Engineering и рефлексивного программирования.

Ключевые слова: интегрированная графическая среда, методический подход, автоматизация программирования, Model-Driven Engineering, метапрограммирование, рефлексивное программирование, онтология, трубопроводная система, энергетика

Введение

Современные трубопроводные системы энергетики (ТПС) (тепло-, водо-, нефте-, газоснабжения и др.) представляют собой уникальные по масштабам и сложности инженерные сооружения, обладающие все возрастающим значением в энергетике, экономике, промышленности, коммунально-бытовом секторе и других сферах жизни страны и общества. Эффективное решение задач их проектирования, реконструкции, эксплуатации и диспетчерского управления невозможно без применения интегрированной графической среды. Эта среда представляет собой программную систему, позволяющую решать информационные (работа с компьютерной моделью ТПС и ее элементами на плане местности в интерактивном режиме), расчетные (решение инженерных задач) и аналитические (просмотр графиков, таблиц и результатов расчетов) задачи в рамках единого интерфейса пользователя. Современная среда должна обеспечивать работу в режиме многоуровневого моделирования, когда моделируемая ТПС может состоять из множества расчетных схем различных типов (схемы сетей, насосных станций, источников и т. д.), и элемент ТПС (например, источник) может быть представлен как элементом расчетной схемы, так и собственной расчетной схемой.

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

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

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

Описание проблемы и постановка задач исследования

Компьютерное моделирование ТПС начинается с построения ее модели, которая представляет собой совокупность структур данных, задающих конфигурацию ТПС, описывает состав ее оборудования и его характеристики, состояние элементов и их свойства (технические характеристики, гидравлические параметры и граничные условия).

ТПС различного типа обладают общностью структурно-топологических свойств и физических законов течения транспортируемой среды [1], что позволяет сформулировать следующие общие положения, характерные для методологии их компьютерного и математического моделирования:

- все ТПС могут быть смоделированы графом, вершины которого соответствуют узлам (источники, узлы-соединения, потребители), а дуги — ветвям (трубопроводы, активные ветви с насосными станциями, регуляторами давления или расхода);
- задачи математического моделирования ТПС обладают общностью своих содержательных и математических постановок, а применяемые для их решения методы, алгоритмы и специализированное программное обеспечение могут носить универсальный характер (не зависеть от типа ТПС);
- компьютерная модель ТПС определенного типа может быть представлена как совокупность графа, описывающего конфигурацию этой системы, и набора графических и математических моделей, описывающих свойства ее элементов.

Сформулированные положения показывают, что существует возможность создания интегрированной графической среды в виде единой универсальной программной системы, которая обеспечит возможность работы пользователя с активными графическими моделями ТПС в интерактивном режиме, когда программная система динамически настраивается на конкретный тип ТПС и состав ее оборудования путем интеграции соответствующих моделей элементов ТПС.

Авторы рассматривали возможность применения объектно-ориентированного и компонентного подходов для построения единой программной системы, позволяющей моделировать различные типы ТПС. Практическое применение этих подходов приводит к возникновению сложной проблемы, вызванной необходимостью интеграции в программную систему большого числа программных компонентов, используемых для моделирования ТПС различных типов, элементы которых могут быть представлены широким набором оборудования и могут находиться в различных состояниях.

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

ТПС определенного типа обладают характерными для этого типа свойствами, составом

прикладных задач и используемым для их решения специализированным ПО. Знания о них необходимо сохранить в форме, пригодной для обработки программными системами, и использовать при решении различных задач. Для этого необходимо организовать хранение этих знаний в виде онтологий [2], которые можно использовать для автоматизации этапов построения ПО, информационного наполнения пользовательского интерфейса и обеспечения эффективной работы с компьютерной моделью ТПС.

Методический подход и его особенности

В результате проведенных авторами исследований предложен методический подход, в рамках которого объединены концепция Model-Driven Engineering ("управляемая моделями разработка", MDE) и рефлексивное программирование.

Концепция MDE представляет собой совокупность методических подходов к автоматизированному построению сложных программных систем на основе предварительно разработанных моделей [3—5]. В рамках предлагаемого подхода MDE адаптирована к особенностям предметной области компьютерного моделирования ТПС, а автоматизированное построение программной системы выполняется на основе компьютерной модели ТПС и онтологий.

Рефлексивное программирование является одним из видов метапрограммирования и представляет собой расширение парадигмы объектно-ориентированного программирования [6—7]. Рефлексивное программирование позволяет выполнять операции, которые в рамках объектно-ориентированного программирования выполнить невозможно, среди которых наиболее важными являются следующие: изучение классов во время выполнения программы, определение их полей, методов и конструкторов; создание новых экземпляров объектов по названию класса с использованием конструкторов; установка и получение значений полей по их названию; вызов методов по их названию и описанию аргументов; гибкая работа с массивами и контейнерами (коллекциями). Рефлексивное программирование позволяет отслеживать структуру программной системы и динамически изменять состав программных компонентов во время ее работы.

Разработанный авторами методический подход включает:

- принципы автоматизированного построения интегрированной графической среды для компьютерного моделирования ТПС;
- архитектуру интегрированной графической среды, описывающую устройство программ-

ной системы, которая создается в автоматизированном режиме;

- методику автоматизированного построения программной системы на основе концепции MDE, рефлексивного программирования, онтологий и компьютерной модели ТПС.

Основные принципы, лежащие в основе методического подхода:

- программная система, ориентированная на моделирование конкретной ТПС, строится в автоматизированном режиме в контексте решения задачи построения компьютерной модели этой ТПС (рис. 1);
- автоматизированное построение программной системы и ее адаптация к конкретным особенностям моделируемой ТПС выполняется средствами рефлексивного программирования;

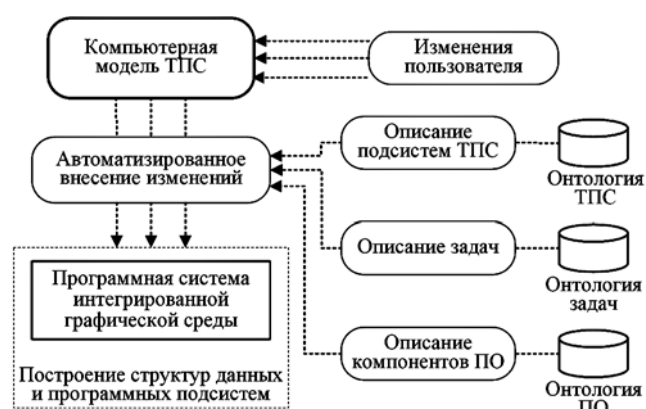


Рис. 1. Иллюстрация принципа построения интегрированной графической среды

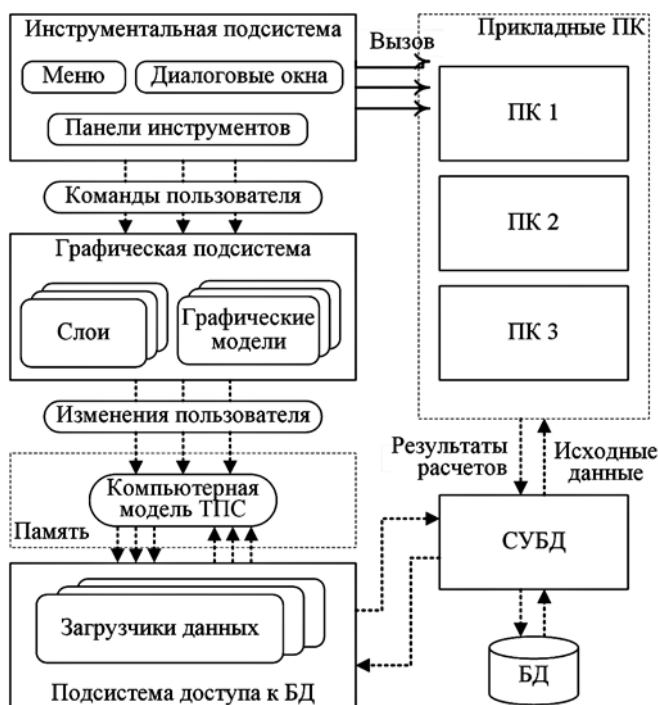


Рис. 2. Архитектура интегрированной графической среды

- для автоматизации этапов построения программной системы и информационного наполнения пользовательского интерфейса используются знания, хранение которых организовано в виде онтологий;
- знания, описывающие общие для всех типов ТПС свойства, задачи и используемое для их решения ПО, сохраняются в виде онтологий, а конкретные особенности моделируемой ТПС описываются ее компьютерной моделью. Авторами предложена архитектура интегрированной графической среды (рис. 2), которая включает следующие составляющие:

1) *графическую подсистему*, которая обеспечивает работу пользователя с активной графической моделью ТПС на плане местности и позволяет ему просматривать данные в удобном для восприятия виде и вносить необходимые изменения;

2) *инструментальную подсистему*, которая обеспечивает решение следующих задач: создает необходимый графический вид меню, позволяя пользователю решать задачи, соответствующие типу моделируемой ТПС; создает и предоставляет пользователю панели инструментов для рисования, редактирования, изменения состояний элементов активной модели ТПС; формирует диалоговые окна для настройки отображения модели ТПС, занесения необходимой информации и параметров, влияющих на решение прикладных задач; создает панели инструментов для рисования элементов плана местности и городской застройки;

3) *подсистему доступа к базам данных*, которая обеспечивает организацию работы с различными базами данных (БД), применяемыми для хранения и многократного использования компьютерных моделей ТПС, исходных данных и результатов расчетов, данных об объектах городской застройки.

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

При реализации интегрированной графической среды в соответствии с предлагаемым методическим подходом авторы ориентируются только на свободное ПО. В качестве базового языка программирования используется Java, выбор которого обусловлен поддержкой современных технологий объектно-ориентированного, компонентного и функционального программирования, а также встроенной поддержкой рефлексивного программирования и других видов метапрограммирования [6, 8]. Для работы с БД используется СУБД Firebird, которая обладает возможностью многопользовательского доступа и переносимостью между различными операционными системами.

Принципы построения онтологий

В рамках интегрированной графической среды используются три онтологии: онтология ТПС, онтология задач и онтология ПО. Для долговременного хранения онтологий авторы отдали предпочтение формату JSON¹, который является универсальным средством обмена данными между программными системами и предоставляет возможность организации хранения сложных (иерархических, сетевых) структур данных, а в современных платформах программирования существуют библиотеки, предназначенные для работы с этим форматом.

Загрузка содержимого онтологий в оперативную память выполняется загрузчиком онтологий (рис. 3), который по типу моделируемой ТПС считывает из файлов JSON соответствующее этому типу содержимое. Для выполнения этой операции используется библиотека GSON². Полученное описание записывается в структуры данных, реализующие представление содержимого онтологий в памяти.

Построение интегрированной графической среды для работы с ТПС конкретного типа осуществляется в автоматизированном режиме на основе описания этого типа систем из онтологии ТПС. Онтология ТПС включает следующие составляющие: описание иерархической структуры построения ТПС различного типа и назначения, классификацию используемого оборудования сетей и его технические характеристики, описание элементов ТПС и их свойств (технических характеристик, гидравлических параметров и граничных условий), классификацию и описание используемых математических моделей (например, уравнения, описывающие законы падения напора на ветвях сети, формулы расчета сопротивлений и др.).

На рис. 4 представлен фрагмент онтологии ТПС. ТПС состоит из элементов, которые подразделяются на узлы и ветви и могут находиться в одном из следующих физических состояний: рабочем, аварийном, отключенном или на реконструкции. Узлы,

в свою очередь, подразделяются на источники, узлы-соединения и потребители, а ветви — на активные и пассивные. Активные ветви содержат оборудование, оказывающее активное влияние на потокораспределение в сети. К ветвям этого типа относятся насосные станции и ветви с регуляторами расхода или давления. Пассивные ветви содержат трубопроводные участки ТПС, которые обладают такими свойствами, как длина и тип прокладки трубопровода. Каждый трубопроводный участок имеет конкретный тип трубопровода, характеризующийся диаметром и материалом. Для трубопровода имеется математическая модель, описывающая законы падения давления, скорость течения теплоносителя и выражение для расчета гидравлического сопротивления, а также графическая модель, определяющая способ отображения.

Онтология задач используется для настройки компьютерной модели ТПС на конкретную задачу и автоматизации процесса подготовки данных для вызова интегрированной графической средой прикладного ПО. Эта онтология содержит описание прикладных задач (например, оптимизации параметров многоконтурной сети) и методов их решения, описание алгоритмов, перечисление параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи.

Онтология ПО предназначена для хранения знаний, необходимых для автоматизации по-

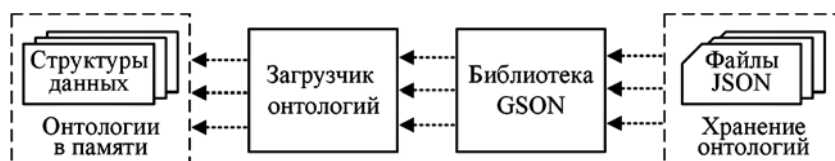


Рис. 3. Иллюстрация процесса загрузки содержимого онтологий

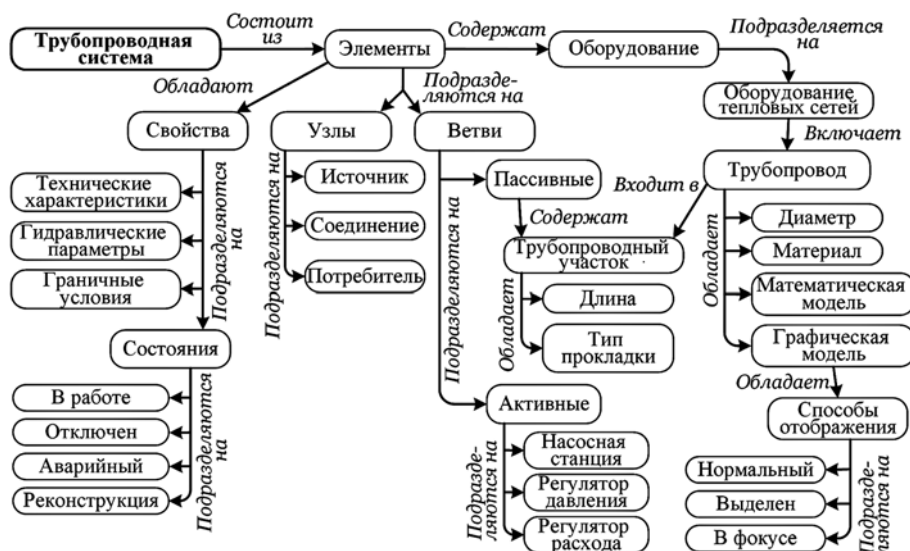


Рис. 4. Фрагмент онтологии трубопроводных систем

¹ <http://json.org/>

² <https://github.com/google/gson>

строения и использования ПО. Эта онтология содержит описание:

- программных комплексов, предназначенных для решения прикладных задач;
- программных компонентов, используемых для построения интегрированной графической среды;
- метаданных (входные и выходные параметры, описание форматов данных);
- технологий и интерфейсов доступа к программным компонентам.

Онтология ПО содержит описание следующих типов программных компонентов:

- компоненты инструментальной подсистемы, используемые для построения элементов пользовательского интерфейса (меню, панели инструментов, диалоговые окна и др.);
- компоненты, используемые для представления отдельных элементов (узлов и ветвей) графа моделируемой ТПС;
- компоненты, реализующие графические модели элементов ТПС определенного типа и в определенном рабочем состоянии;
- компоненты, реализующие структуры данных для описания отдельных элементов конкретных ТПС (технические характеристики, гидравлические параметры и граничные условия);
- компоненты графической подсистемы, используемые для отображения геометрических фигур, формирующих объекты городской застройки на плане местности (линии, полигоны, надписи и др.);
- компоненты доступа к БД, используемые для чтения и записи данных в соответствии с типом ТПС.

Методика автоматизированного построения

В соответствии с предложенной методикой автоматизированное построение интегрированной графической среды для компьютерного моделирования ТПС осуществляется в результате выполнения следующих трех этапов:

- 1) построение инструментальной подсистемы;
- 2) построение подсистемы доступа к базам данных;
- 3) построение графической подсистемы.

На первом и втором этапах выполняется настройка среды на выбранный пользователем тип ТПС, а на третьем этапе — настройка на работу с конкретными особенностями моделируемой ТПС.

Рассмотрим подробнее содержание этих этапов.

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

Построение на первом этапе выполняется в соответствии со следующим алгоритмом.

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

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

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

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

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

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

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

Структура формируемой инструментальной подсистемы представлена на рис. 5.

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

Этап 3. Построение графической подсистемы. Основная идея, предлагаемая авторами, заключается в том, что графическая подсистема, от-



Рис. 5. Структура инструментальной подсистемы

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

Когда пользователь добавляет новый элемент к модели ТПС, то этот элемент отображается в базовом состоянии, например, для ветви это будет трубопровод в активном состоянии. Если пользователь изменит состояние элемента или укажет установленное на этой ветви оборудование, то среда автоматически подключит соответствующую модель элемента и отобразит ее в составе компьютерной модели ТПС. Например, если пользователь укажет, что на ветви установлен регулятор давления типа "до себя", то соответствующий компонент-модель из библиотеки компонентов будет интегрирован в программную систему.

Каждый элемент моделируемой ТПС представляется отдельным программным компонентом и обладает уникальным номером. Все компоненты, реализующие узлы (источники, узлы-соединения, потребители), являются экземплярами класса *Node*, все компоненты, реализующие ветви (трубопроводы, насосные станции, ветви с регуляторами давления и расхода) — экземплярами класса *Branch*. Хранение в памяти этих компонентов организовано с помощью хэш-таблиц. Схема процесса построения графической подсистемы представлена на рис. 6.

Когда пользователь создает новый элемент компьютерной модели ТПС, менеджер моделей

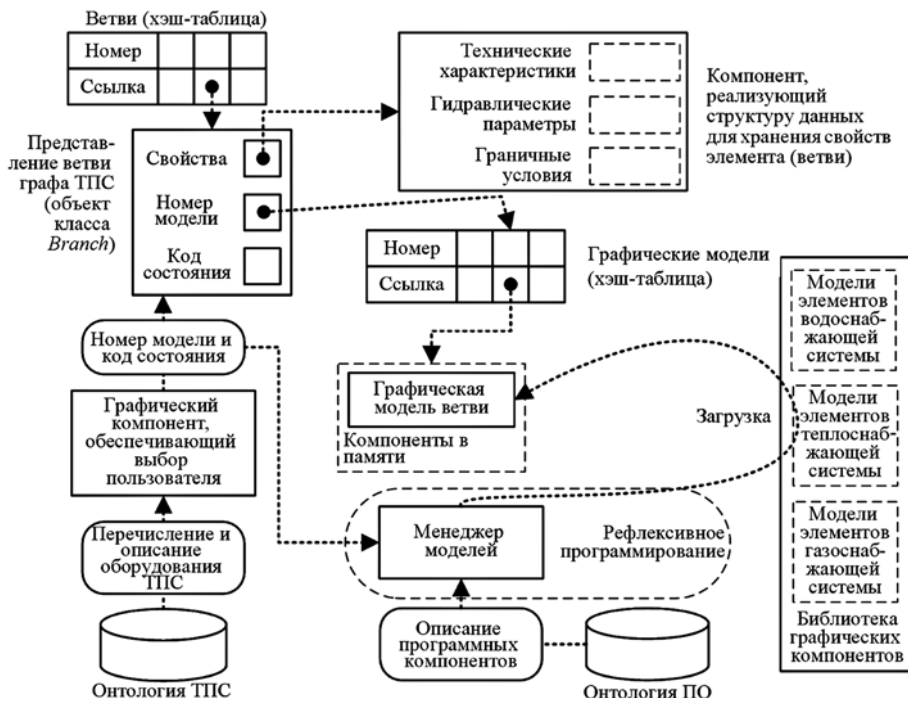


Рис. 6. Схема процесса построения графической подсистемы

выделяет память для хранения структуры данных, описывающей этот элемент. Далее пользователь заносит данные о свойствах этого элемента ТПС через соответствующие средства графического интерфейса. После этого пользователь выбирает модель, описывающую характеристики оборудования, и устанавливает состояние элемента. Менеджер моделей проверяет, подключен ли необходимый компонент-модель, реализующий выбранную пользователем модель элемента ТПС. Если необходимая модель не является доступной, то она интегрируется в программную систему из библиотеки графических компонентов средствами рефлексивного программирования. Когда модель становится доступной, то в структуру данных, описывающую новый элемент ТПС, заносится номер этой модели. Однажды подключенный к программной системе компонент-модель может быть многократно использован для представления элементов ТПС, тип и состояние которых соответствует этой модели.

Практическая реализация

Разработанный методический подход применен при реализации программной системы, которая представляет собой программный прототип интегрированной графической среды для компьютерного моделирования ТПС. Реализованный программный прототип ориентирован на работу с компьютерными моделями теплоснабжающих

систем и позволяет создавать, вносить изменения и сохранять их в БД для многократного использования при решении прикладных задач.

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

На рис. 7 представлен вид главного окна интегрированной графической среды, которое содержит элементы интерфейса, формируемые инструментальной и графической подсистемами. Инструментальная подсистема обеспечивает работу следующих элементов главного окна интегрированной графической среды:

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

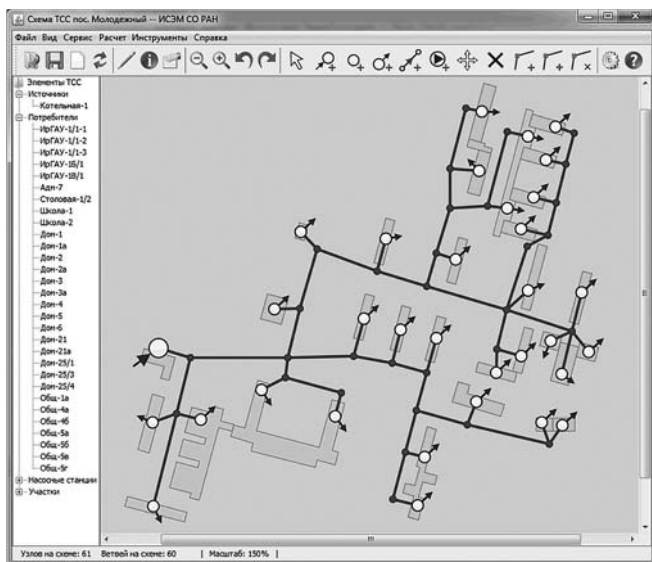


Рис. 7. Вид графического интерфейса интегрированной среды

Центральную часть главного окна занимает рабочая область, отвечающая за работу пользователя с активной графической моделью ТПС и объектами городской застройки на плане местности. За реализацию этой части пользовательского интерфейса отвечает графическая подсистема, которая динамически адаптируется (формируется) к следующим действиям пользователя:

- загрузка в память данных, описывающих ранее созданную модель ТПС;
- создание новой модели ТПС при выборе пользователем ее типа;
- изменения, вносимые пользователем в компьютерную модель ТПС.

Заключение

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

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

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

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

Список литературы

1. Меренков А. П., Хасилев В. Я. Теория гидравлических цепей. М.: Наука, 1985. 280 с.
2. Коголовский М. Р. Перспективные технологии информационных систем. М.: ДМК-Пресс, 2003. 288 с.
3. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering #1. USA, Morgan & Claypool, 2012. 182 p.
4. Völter M., Stahl T., Bettin J., Haase A., Helsen S. Model-Driven Software Development: Technology, Engineering, Management. USA, Wiley, 2006. 446 p.

5. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение концепции Model-Driven Engineering в программном комплексе для определения оптимальных параметров теплоснабжающих систем // Программная инженерия. 2016. Т. 7, № 3. С. 108—116.

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

7. Стенников В. А., Барахтенко 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.

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

V. A. Stennikov, Corresponding Member of the Russian Academy of Sciences,

Professor, Honored Science Worker of the Russian Federation, Director, e-mail: sva@isem.irk.ru,

E. A. Barakhtenko, Ph. D. in Technical Sciences, Senior Researcher, e-mail: barakhtenko@isem.irk.ru,

D. V. Sokolov, Ph. D. in Technical Sciences, Senior Researcher, e-mail: sokolov_dv@isem.irk.ru,

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

A Methodology for the Creation of an Integrated Graphical Environment for Computer Modeling of Energy Pipeline Systems

This paper presents a new methodology for the creation of an integrated graphical environment for computer modeling of energy pipeline systems of various types. This environment is a software system for solving information, computational and analytical problems within a single user interface.

The developed methodology includes:

- principles of automated creation of an integrated graphical environment for computer modeling of energy pipeline systems;
- architecture of the integrated graphical environment, describing the design of the software system created in an automated mode;
- procedure for automated software system creation based on the Model-Driven Engineering paradigm, reflective programming, ontologies and computer model of the pipeline system.

The main ideas underlying the methodology are:

- The software system focused on modeling a particular pipeline system is created in an automated mode in the context of solving the problem of building a computer model of this pipeline system.
- The automated construction of the software system and its adaptation to specific features of the modeled pipeline system is performed on the basis of reflective programming.
- The knowledge describing common properties of all pipeline system types, problems and software, are stored as ontologies, and specific features of the modeled pipeline system are described by its computer model.
- The automation of stages of the software system creation, and design of the user interface, involves the knowledge stored as ontologies.

The implemented software system is a software prototype of an integrated graphical environment for computer modeling of energy pipeline systems.

The proposed methodology is universal and can be used in organizations engaged in the development of software for computer modeling of energy pipeline systems of various types and purposes.

Keywords: integrated graphical environment, methodology, automation of programming, Model-Driven Engineering, metaprogramming, reflective programming, ontology, pipeline system, power engineering, software engineering

References

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

2. Kogalovskii M. R. *Perspektivnye tekhnologii informatsionnykh system* (Perspective Technologies of Information Systems), Moscow, DMK-Press, 2003, 288 p. (in Russian).

3. Brambilla M., Cabot J., Wimmer M. *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering No. 1*. San Rafael, Morgan & Claypool, 2012, 182 p.

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

5. Stennikov V. A., Barakhtenko E. A., Sokolov D. V. *Primenenie kontseptsii Model-Driven Engineering v programmnom*

komplekse dlya opredeleniya optimal'nykh parametrov teplosnabzhayushchikh sistem (Model-Driven Engineering in the Software for Parameter Optimization of Heat Supply Systems), *Programmnyaya Inzheneriya*, 2016, vol. 7, no. 3, pp. 108—116 (in Russian).

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

7. Stennikov V. A., Barakhtenko E. A., Sokolov D. V. *Primenenie metaprogramirovaniya v programmnom komplekse dlya resheniya zadach shemno-parametricheskoy optimizatsii teplosnabzhayushchikh sistem* (Metaprogramming in the software for solving the problems of heat supply system schematic and parametric optimization), *Programmnyaya Inzheneriya*, 2011, no. 6, pp. 31—35 (in Russian).

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