ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ИНСТРУМЕНТАЛЬНАЯ СРЕДА ДЛЯ РАЗРАБОТКИ СИСТЕМ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ

В.А. Семенов, С.В. Морозов,
О.А. Тарлапан, Е.Ю. Ширяева

В статье описывается объектно-ориентированная инструментальная среда программирования, предназначенная для унифицированной разработки систем численного моделирования в различных предметных областях. Среда реализует обобщенную концепцию моделирования, подобную диакоптике Крона, и представляет собой систему классов объектов, выражающих основные ее понятия. Среда реализована на языке Си++. Рассматриваются примеры разработки систем параметрического геометрического моделирования и моделирования течений в сетях трубопроводов с использованием предложенной среды.

1. Введение

В настоящее время численное моделирование с успехом применяется при анализе сложных физических явлений и проектировании разнообразных технических систем. Несмотря на имеющееся понимание общности в математических постановках задач и вычислительных подходах к их решению, до сих пор программы моделирования разрабатываются отдельно для каждой предметной области, а иногда и для каждого конкретного класса задач. При этом необходимость создания специализированных программ обычно мотивируется наличием особенностей в постановках задач, существованием частных аналитических решений, спецификой применения численных методов и связанными с ними различными способами программной реализации. Проблемная и процедурная ориентация технологий программирования в значительной степени способствовали такой специализации.

С развитием объектно-ориентированных технологий, предоставляющих мощные инструментальные возможности для множественного использования программного обеспечения, попытки унифицированной разработки систем численного моделирования в различных предметных областях кажутся естественными и вполне реализуемыми. Создание приложений с использованием объектно-ориентированных математических библиотек, графических интерфейсов, баз данных является в настоящее время самой привычной практикой [1–3].

Главное препятствие на этом пути возникает в связи с необходимостью унификации программных средств постановки задач, которые существенно зависят от особенностей прикладной задачи и специфики предметной области, в которой данная задача возникает. Разработка или использование какой-либо обобщенной методологии моделирования в данном случае позволили бы формализовать этапы постановки и решения задач моделирования, хотя бы в самом общем виде, и специфицировать необходимые базовые инструментальные средства разработки приложений. Несмотря на отсутствие единой концепции численного моделирования, привлечение идей диакоптики Крона [4, 5] является на наш взгляд многообещающим и позволяет в значительной степени продвинуть решение данной проблемы.

В настоящей работе рассматриваются вопросы проектирования и разработки объектно-ориентированной инструментальной среды для численного моделирования нелинейных статических систем. Рассматриваемый класс физических систем математически описывается нелинейными системами алгебраических уравнений и охватывает довольно широкую область моделирования, включающую электрические схемы, механические, гидравлические, энергетические системы, нейронные сети, дискретные варианты многочисленных задач математической физики. Описываемая концепция моделирования и соответствующая инструментальная среда допускают непосредственное обобщение на случай нелинейных динамических стохастических систем.

2. Концептуальная модель физической системы

Предлагается следующая концепция численного моделирования, которая фактически определяет способ формирования и представления обобщенной математической модели различных физических систем. В определенном смысле концепция следует диакоптике Крона — науке об исследовании сложных систем по частям, которая получила определенное практическое распространение, в частности, при моделировании больших интегральных схем [6].

Будем считать, что моделируемая физическая система $S = \{K, C\}$ является системой взаимосвязанных компонентов и представляется конечным множеством компонентов $K_i, i = 1,\ldots,n$, и их связей $C_j, j = 1,\ldots,m$. Компонентом системы может быть либо элемент, либо подсистема, состоящая из элементов или подсистем более низкого уровня. Тем самым допускается, что рассматриваемая система может иметь сложную многоуровневую иерархическую структуру.

Каждый компонент K_i имеет определенное число внешних соединений $L_{ik}, k = 1,\ldots,k_i$, через которые осуществляется его связь с другими компонентами в системе. Каждая связь C_j при этом объединяет произвольное число соединений компонентов $C_j = \left\{L_{ik}, i = i_j(l), k = k_j(l), l = 1,\ldots,l_j\right\}$. Для определенности будем считать, что каждое соединение участвует только в одной связи, в каждой связи участвует по крайней мере два соединения, у компонентов системы отсутствуют несвязанные соединения и, как следствие, выполняется $\sum_{i=1}^{n} k_i = \sum_{j=1}^{m} l_j$.

Любой компонент–подсистема может рассматриваться как самостоятельная система, состоящая из своих собственных компонентов и их связей, но имеющая определенное число внешних соединений, благодаря которым она может быть включена в систему более высокого уровня. Поскольку любая подсистема состоит из элементов и подсистем более низкого уровня, исходная система всегда может быть преобразована в эквивалентное представление в виде взаимосвязанных элементов. В дальнейшем будем рассматривать элементные представления $S = \{E, C\}$.

Каждый элемент системы E_i классифицируется в соответствии с видом математической модели и, возможно, со значениями ее параметров p_i. Моделью элемента являются алгебраические соотношения, связывающие внутреннее состояние элемента x_i со значениями переменных его внешних связей y_i и параметрами модели p_i. В самом общем случае соотношения представляются неявно в виде нелинейных алгебраических уравнений $F_i(x_i, y_i, p_i) = 0$, где $F_i : R^{n_i} \to R^{m_i}$, $n_i = n_i^x + n_i^y + n_i^p$.

Вектор переменных связей отдельного элемента y_i собирается из отдельных компонентов z, соответствующих связям системы, которыми соединен данный элемент. При этом вектор z определяется непосредственно на множестве связей системы, причем с каждой связью C_j обычно связывается свое подмножество компонентов $z_j \in R^{nz}$ одного и того же фиксированного размера. Таким образом, если соединение L_{ik} элемента E_i участвует в связи C_j, то $y_{ik} = z_j$ и $n_i^y = l_i\, nz$.

Иногда при $n_i^x = m_i$ удается явно разрешить уравнения относительно переменных состояния элемента и представить модель элемента в виде соотношений $x_i = \Phi_i^x(y_i, p_i)$. В ряде случаев при $n_i^y = m_i$ уравнения разрешаются относительно переменных связей и представляются соотношениями $y_i = \Phi_i^y(x_i, p_i)$. Не менее содержательным оказывается случай $n_i^y > m_i$, когда уравнений не достаточно для однозначного определения всех переменных связей, однако возможно задание набора различных правил вида $y_i' = \Phi_i^y(x_i, y_i'', p_i)$, применимых для вычисления одного подмножества переменных y_i' относительно другого y_i''. В конечном счете способ задания и представления модели элемента связан со спецификой задачи. Мы же будем считать, что в рассматриваемом классе задач модели элементов всегда могут быть заданы по крайней мере одним из перечисленных способов.

Обсудим вопрос формирования модели всей системы. Прежде всего заметим, что параметрами системы $p = \{p_1, p_2, ..., p_n\}$ являются параметры моделей отдельных элементов. Состояние всей системы определяется состоянием всех элементов и вектором переменных связей, поэтому вектор состояния формируется как $x = \{x_1, x_2, ..., x_n, y\}$. Поведение всей системы описывается некоторой системой нелинейных алгебраических уравнений $F(x, p) = 0$, которая и является ее математической моделью. Главное отличие математической модели системы от компонентных моделей состоит в том, что переменные внутренних связей вошли в вектор состояния, а внешние связи у замкнутой системы отсутствуют.

К сожалению, в различных прикладных задачах построение моделей осуществляется специфическим образом, в силу различий применяемых фундаментальных законов и особенностей элементных моделей. Не претендуя на какую-либо всеобъемлемость, тем не менее, можно попытаться выделить достаточно общий и распространенный в различных предметных областях подход к формированию моделей.

Прежде всего в систему уравнений должны быть включены элементные уравнения $F_i(x_i, y_i, p_i) = 0$, $i = 1, \ldots, n$, выражающие функциональные свойства отдельных элементов. При этом элементные уравнения могут либо явно включаться в полную систему уравнений, либо учитываться неявно при формировании других уравнений системы с использованием упомянутых специальных представлений моделей. Последние позволяют выразить одни переменные через другие и тем самым формировать оставшиеся уравнения в сокращенном базисе переменных.

Кроме элементных уравнений, в систему могут быть включены топологические уравнения, отражающие способ вхождения отдельных компонентов в моделируемую систему. Довольно часто в приложениях топологические уравнения выражают фундаментальные физические законы сохранения величин, соответствующих переменным состояния. В таких случаях переменные состояния обычно задают потоки сохраняемых величин через соединения элементов и топологические уравнения могут быть представлены в аддитивной алгебраической форме. Уместны аналогии с законами Кирхгофа в теории цепей и с консервативными разностными методами для уравнений математической физики. Таким образом, каждой связи C_j, $j = 1, \ldots, m$ могут быть приписаны уравнения сохранения вида $\sum_{l=1}^{l_j} y_{i\,k} = 0$, в которых индексы $i = i_j(l), k = k_j(l)$ перечисляют все соединения элементов, участвующих в связи.

Ввиду произвольности элементных моделей и топологии всей системы, решение нелинейных уравнений должно осуществляться численным образом с использованием известных численных и, возможно, алгебраических методов [7, 8]. Описанная концептуальная схема является достаточно общей и может быть применена к самым широким классам задач моделирования, включая разнообразные технические системы, инженерные сети, конечно-элементные конструкции.

Концепция допускает дальнейшее обобщение на случай нелинейных динамических стохастических систем. В последнем случае математическая формализация приводит к нелинейным системам обыкновенных дифференциальных уравнений со случайно распределенными параметрами.

3. Объектная модель физической системы

Объектно-ориентированный подход рассматривается нами как подход, применение которого обеспечивает необходимую функциональность и инструментальность разрабатываемой программной среды. Неформальное использование ООП предполагает проведение объектного анализа предметной области и построение ее объектной модели, определяющей основные виды объектов и способы их взаимодействия в рамках приложений.

Предлагаемая среда программирования представляет собой систему классов объектов, выражающих ключевые понятия рассмотренной выше концепции моделирования, а именно: ***System***, ***Component***, ***Element***, ***Subsystem***, ***Connection***, ***Link***, ***Simulation***, ***Model***, ***Analysis***. Данное базовое множество вместе с классами математической библиотеки является достаточным для представления разнообразных физических систем и реализации наиболее общих методов постановки и решения задач моделирования в рамках описанной концепции.

Абстрактный класс ***Component*** реализует понятие компонента системы. С компонентом связывается набор атрибутов, включающий идентификатор компонента, параметры, переменные состояния и внешние соединения. Параметры и переменные состояния используются при анализе моделируемой системы. Соединения определяют потенциальную возможность топологического включения компонента в систему и фактически определяют способ взаимодействия компонентов в системе. Для объектов данного класса определены методы:

- задания и получения идентификатора компонента,

- оперирования с параметрами (определения их количества, доступа к ним),
- оперирования с переменными состояния,
- редактирования связей компонента в системе,
- представления статической, динамической и частотной характеристик компонента (функциональных отображений соответствующих типов).

Для каждого параметра компонента определено символическое имя, значение, а также единица и масштаб измерения. Каждому соединению компонента может быть приписан статус, определяющий соединение как входное, выходное или неопределенное.

Включение компонента в систему осуществляется в результате задания связей компонента через массив соединений класса ***Link***, которые представляют собой ссылки на объекты класса ***Connection***. Каждая связь имеет идентификатор, вектор переменных и в свою очередь может содержать список соединений, которые она объединяет. Информация о статусе соединений может использоваться для контроля топологической корректности описания всей системы. Например, часто при моделировании логических схем исходят из предположения, что в каждой связи участвует лишь один выход и произвольное или ограниченное число входов. С этой целью может оказаться полезным классифицировать связи в соответствии с реализуемыми правилами контроля топологической корректности.

Классы ***Element***, ***Subsystem*** конкретизируют понятие компонента системы как элемента или подсистемы и определяются как производные от класса ***Component***. В отличие от элементов, подсистемы хранят ссылку на соответствующее им представление класса ***System***. Тем самым подсистемы, участвуя в описании моделируемой системы, рассматриваются и как ее компоненты, и как самостоятельные системы со своими отдельными представлениями.

Подобный дуализм обеспечивает возможность применения как традиционных, так и диакоптических методов моделирования. В первом случае исходное представление системы должно быть преобразовано в эквивалентное ему элементное представление в результате рекурсивного раскрытия подсистем до элементного уровня. Во втором случае такие преобразования не обязательны и вполне допустимо представление моделируемой системы на уровне подсистем (см. рис. 1).

Класс ***Element*** является базовым для конкретных классов моделей элементов, участвующих в приложении. Обычно число таких моделей велико и классификация элементов приобретает развернутый вид. Например, при проектировании электронных устройств используется стандартный набор элементов, включающий различные зависимые и независимые источники, резисторы, конденсаторы, индуктивности, транзисторы, причем для большинства типов элементов применяются разнообразные математические модели со своими специфическими функциональными характеристиками и наборами параметров.

Класс ***System*** поддерживает представление моделируемой системы в виде множеств компонентов и их связей (см. рис. 1). Класс определяет и реализует методы описания и редактирования системы, включая создание, удаление, поиск, перестановку компонентов и их связей.

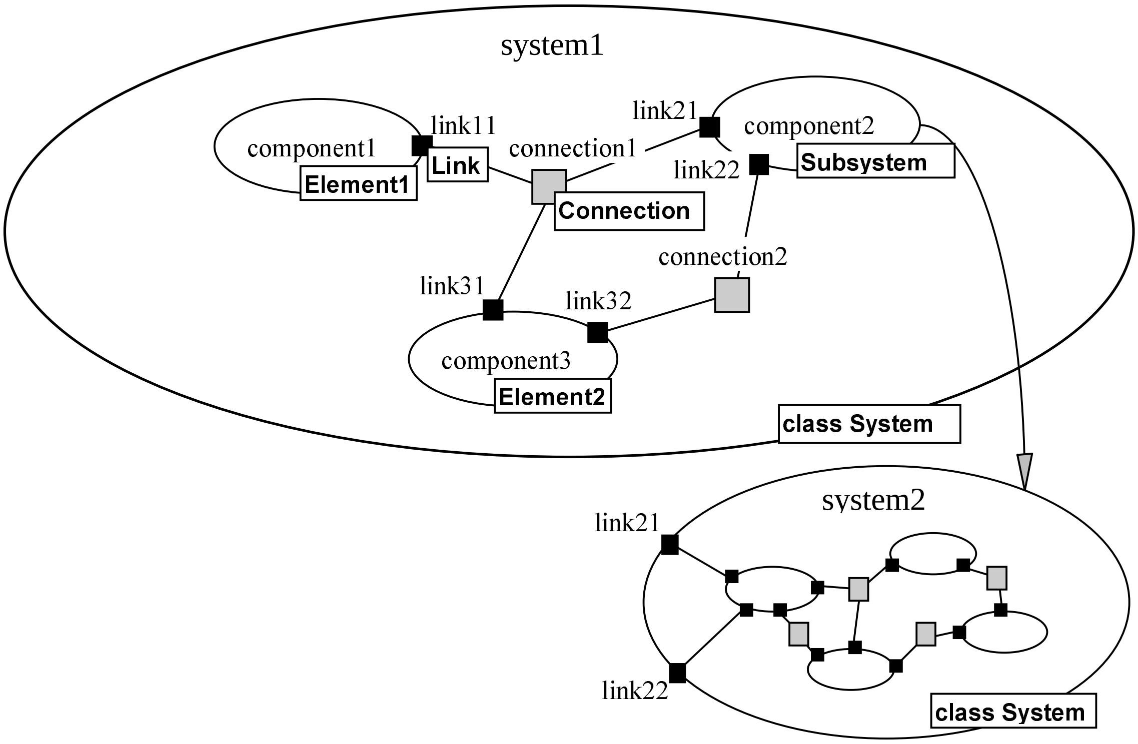

Рис. 1. Диакоптическое представление моделируемой системы

Класс ***Analysis*** предназначен для постановки задач моделирования, которая включает указание вида анализа, задание начальных условий или начального приближения, варьируемых параметров, а также указание искомых переменных, которые подлежат постобработке и выводу пользователю. Параметры могут задаваться в режиме простого, многовариантного или статистического расчета, сочетающемся с любым из основных видов анализа. Специальные производные классы ***StaticAnalysis***, ***DynamicAnalysis***, ***FrequencyAnalysis*** уточняют возможные виды анализа как статический, динамический и частотный, определяя необходимые дополнительные параметры моделирования, например, интервал времени, частотный диапазон.

Заметим, что рассмотренная система классов в значительной степени отражает прикладную сторону моделирования и по существу определяет способ унифицированной постановки задач моделирования в терминах пользователя. Другой, не менее содержательной стороной моделирования является математическая формализация прикладной задачи и ее решение одним из вычислительных методов. Проведение статического, динамического и частотного моделирования связано с решением трех основных видов вычислительных задач, к которым сводятся данные виды анализа: нелинейных систем алгебраических уравнений, задач Коши для нелинейных систем обыкновенных дифференциальных уравнений и линейных систем алгебраических уравнений.

Для решения возникающих математических задач целесообразно использовать математическую объектно-ориентированную библиотеку общего назначения [9–14], предоставляющую необходимые инструментальные и алгоритмические средства. Библиотека представляет собой систему наследуемых математических, проблемных и алгоритмических классов. Перечислим лишь некоторые из них, которые имеют непосредственное отношение к обсуждаемому кругу задач.

Математические классы ***Vector***, ***Matrix***, ***Function*** реализуют понятия вектора, матрицы и функции. Для объектов данных классов определены соответствующие базовые операции линейной алгебры и анализа. Проблемные классы ***LinearSystem***, ***NonlinearSystem*** и ***CauchyProblem*** реализуют методы постановки соответствующих задач, задания алгоритмов их решения и получения результатов. Алгоритмические классы ***LinearSystemAlgorithm***, ***NonlinearSystemAlgorithm*** и ***CauchyProblemAlgorithm*** осуществляют непосредственное численное решение задач. Более подробные сведения о принципах построения и основных компонентах библиотеки можно найти в [9–14].

Для того чтобы воспользоваться средствами математической библиотеки, необходимо представить прикладную задачу в стандартной математической форме с использованием ее базовых классов. Будем считать, что классы ***StaticModel***, ***DynamicModel***, ***FrequencyModel***, производные от ***Model***, обеспечивают представление соответствующих статических, динамических и частотных математических моделей в требуемой форме. Организация данной классификации связана с тем, что каждая физическая система может моделироваться различным образом в соответствии с видом проводимого анализа и способом формирования математической модели. Базовый класс ***Model*** обеспечивает необходимый доступ к представлению моделируемой системы типа ***System*** через соответствующую инкапсулируемую ссылку.

Обсудим ключевой для настоящего проекта вопрос о формировании математических моделей физических систем. Ограничимся случаем статического анализа, в котором модель представляется системой нелинейных алгебраических уравнений. Задание уравнений в математической библиотеке осуществляется процедурным образом в результате конструирования векторной функции нескольких переменных класса ***VectorMultiVariate-Function***, для которой определены необходимые операции вычисления значений функции и ее матрицы Якоби. С этой целью класс ***StaticModel*** строится в результате множественного наследования от ***Model*** и ***VectorMultiVariateFunction***, в результате чего он получает доступ к представлению моделируемой системы и приобретает необходимый интерфейс функционального объекта. Реализация вычислительных операций осуществляется путем итерирования элементов системы и расчета аддитивных вкладов отдельных элементов в выражения функции и матрицы Якоби с использованием их статических характеристик. Сами операции реализуются в конкретных наследуемых классах ***StaticModel1***, ***StaticModel2*** и ***StaticModel3***, которые соответствуют трем наиболее распространенным в приложениях способам формирования статических моделей в рамках обсуждаемой концепции.

Первый способ соответствует наиболее простому случаю, когда статическая модель собирается из отдельных элементных уравнений. Данным способом происходит, например, формирование параметрической геометрической модели из отдельных уравнений, связанных с множеством наложенных ограничений [16].

Во втором способе, кроме элементных уравнений, в модель добавляются топологические уравнения, причем компоновка уравнений осуществляется в полном базисе переменных состояния и переменных связей. Характерными примерами могут служить известные гибридные представления уравнений электронных схем и течений в сетях трубопроводов [19, 20].

Наконец, третий способ соответствует частному, но распространенному случаю, когда элементные уравнения могут быть явно разрешены относительно переменных состояния и модель физической системы составляется из топологических уравнений, в которых переменные состояния выражены через переменные связей на основе элементных уравнений. Этому способу соответствуют частные узловые представления в упомянутых выше примерах моделирования электронных схем и течений в трубопроводах.

Основным классом, соответствующим понятию самого программного приложения и объединяющим все рассмотренные выше средства численного моделирования, является ***Simulation***. Класс реализует методы:

• задания моделируемой системы типа ***System*** и доступа к ней,

• задания статической, динамической и частотной моделей соответствующих типов ***StaticModel***, ***DynamicModel***, ***FrequencyModel***,

• задания методов решения стандартных математических задач типов ***NonlinearSystemAlgorithm***, ***CauchyProblemAlgorithm***, ***LinearSystemAlgorithm***,

• задания условий моделирования в виде задач одного из типов ***StaticAnalysis***, ***DynamicAnalysis***, ***FrequencyAnalysis***,

• проведения моделирования,

• постобработки результатов и вывода их пользователю.

4. Принципы создания приложений с использованием среды

Описанная система классов позволяет ставить и решать задачи моделирования в самом общем унифицированном виде без какой-либо конкретизации предметной области моделирования. Создание конкретного приложения на основе данной среды сводится в основном к разработке специализированной библиотеки моделей элементов, в результате которой уточняются типы элементов, присущие конкретной предметной области, а также их математические свойства. Библиотека моделей реализуется в виде системы родственных классов, производных от базового класса ***Element***.

В некоторых случаях может потребоваться разработка специальных классов моделей типа ***Model***, необходимых для уточнения, в чем состоит математическая постановка задачи моделирования и каким именно способом следует формировать модель физической системы. Мы полагаем, что число таких способов невелико и они всегда могут быть реализованы в виде стандартной библиотеки классов. Для задач статического моделирования мы выделили три наиболее общих подхода к формированию моделей. Тем не менее, мы не исключаем возможности существования прикладных задач, которые потребуют иного способа математической формализации и приведут к необходимости расширения этого набора.

При создании законченных пользовательских систем данная среда легко интегрируется с соответствующими средствами интерактивной графики и базами данных (см. рис. 2). Для этого выделенным базовым понятиям физической системы и элемента следует придать дополнительный графический и информационный смысл и создать соответствующие производные классы с расширенным набором функций. Как показал опыт практических разработок, в ряде случаев целесообразно надстроить среду моделирования на имеющейся графической библиотеке, поскольку при этом удается унифицировать не только средства моделирования, но и средства пользовательского оконного интерфейса и графического редактора.

Итак, перечислим основные инструментальные возможности разработанной объектно-ориентированной среды моделирования.

• *Предметная адаптация среды* является ее ключевой функциональной особенностью. Разработка приложения сводится к построению производных классов и реализации относительно небольшой части методов. При этом необходимые добавления и изменения носят локальный характер и не нарушают целостность всей среды программирования. В конечном итоге это обеспечивает минимальные затраты на разработку новых программных приложений, связанных с численным моделированием, и возможность создания целых линий программных продуктов.

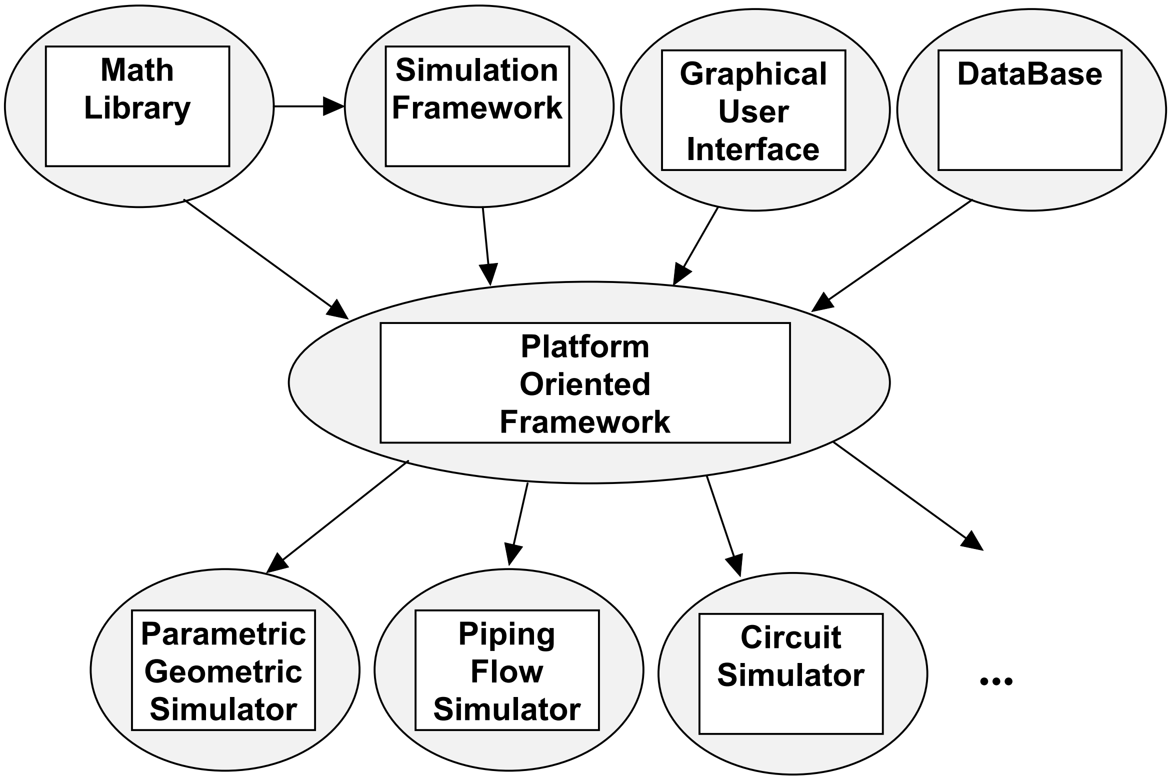

Рис. 2. Инструментальные средства разработки приложений моделирования

• *Проблемная адаптация среды* заключается в возможности введения новых видов анализа и обеспечивается базовым классом ***Analysis***, определяющим основные средства задания условий моделирования и постановки математической задачи.

• *Задание новых способов формирования модели системы* осуществляется в результате конструирования класса, производного от ***Model***, хотя в большинстве случаев удается воспользоваться уже реализованными стандартными вариантами формирования модели.

• *Включение новых видов элементов* обеспечивается организацией библиотеки моделей элементов и сводится к построению производных классов на основе базового класса ***Element***.

• *Алгоритмическая адаптация среды* является наиболее сложным и важным аспектом ее разработки и применения, поскольку любая попытка создания универсального математического и прикладного программного обеспечения может приводить к крайне низкой его эффективности при решении конкретных частных классов задач. Решение этой проблемы мы связываем с использованием математической объектно-ориентированной библиотеки, предоставляющей мощные инструментальные возможности для развития проблемного и алгоритмического репертуара и его адаптации к конкретным прикладным проблемам [9–14].

5. Система параметрического геометрического моделирования

Программные средства параметрического геометрического моделирования составляют неотъемлемую часть современных развитых CAD/CAM систем. Вообще говоря, термин “*параметрическое моделирование*” включает в себя самые общие методики конструкторского проектирования [15]. В настоящей статье данному термину мы придаем более узкое значение, а именно — моделирование с использованием развитого набора геометрических, размерных и алгебраических ограничений для произвольных геометрических сцен. Последние могут быть представлены каркасной, твердотельной или граничной моделью.

Существуют два основных подхода к решению задач параметрического геометрического моделирования: численный и алгебраический. Численный подход следует наиболее общей концепции параметрического моделирования, ориентированной на обобщенную постановку широкого класса задач параметризации в виде нелинейных систем алгебраических уравнений и применение соответствующих вычислительных методов для их решения [16]. Алгебраический подход основывается на анализе априорных зависимостей между параметрами геометрической модели и последовательном вычислении значений переменных на основе иерархии правил, выражающих одни переменные через другие [17]. Алгебраические методы выигрывают в точности и скорости у численных методов, однако проигрывают им в универсальности. Круг решаемых с их помощью задач ограничен иерархиями ограничений, не включающих циклы и совместные уравнения.

Рассмотрим систему параметрического моделирования, созданную на основе объектно-ориентированной инструментальной среды.

5.1. Постановка и методы решения задач параметризации

Параметрическая модель представляет собой конечное множество ограничений, заданных на множестве геометрических примитивов сцены. Заметим, что все геометрические, размерные и алгебраические ограничения можно редуцировать к ограниченному набору элементарных ограничений, которые определяются только на характерных точках геометрических объектов. Таким образом, множество связей параметрических элементов составляют характерные точки примитивов.

Моделью ограничения являются простейшие алгебраические соотношения, связывающие значения координат его точек с переменными и постоянными значениями его параметров. Будем считать, что геометрические примитивы сами по себе не имеют математических моделей, а в случае необходимости неявно вносят соответствующие ограничения в параметрическую модель сцены. Например, параллелепипед дополнительно определяет 15 ограничений: 3 перпендикулярности и 12 равенств проекций противолежащих ребер. На уровне формирования параметрической модели и ее решения не имеет смысла различать ограничения, заданные пользователем явно, и ограничения, порождаемые геометрическими примитивами, поскольку и те, и другие имеют одинаковые математические модели.

Система нелинейных алгебраических уравнений, являющаяся математической моделью параметризованной сцены, формируется из уравнений отдельных ограничений в базисе переменных, включающем координаты характерных точек и другие параметры геометрической модели. Особенность задачи параметрического геометрического моделирования заключается в том, что в общем случае система уравнений получается недоопределенной или переопределенной. Для решения систем применяются методы ньютоновского типа [18], обеспечивающие надежное решение уравнений в подобных ситуациях.

5.2. Специализированная библиотека элементов параметрического моделирования

Рассмотрим организацию специализированной библиотеки параметрических элементов в рамках объектно-ориентированной инструментальной среды. Библиотека включает основные виды геометрических примитивов и ограничений, участвующих в параметрическом моделировании, и представляется иерархией классов, наследуемых от базового класса ***Element*** (см. рис. 3).

Элементами параметрического моделирования являются прежде всего геометрические примитивы конструктивной твердотельной геометрии, граничного представления и каркасного моделирования, а также элементарные геометрические, размерные и алгебраические ограничения, которые связывают координаты характерных точек геометрической модели и введенные переменные между собой. Существенным для проведения параметрического моделирования оказывается определение дополнительных вспомогательных примитивов — конструкционных элементов, которые неявно участвуют в построении геометрических моделей, задавая систему ограничений для примитивов основных видов. Хотя данные элементы повторяют основные геометрические объекты, они выполняют совершенно иные функции в проведении параметрического моделирования. Обычно средства их построения, визуализации и оперирования с ними отличаются от средств работы с эквивалентными геометрическими примитивами, поэтому мы рассматриваем их в качестве самостоятельных объектов.

- **Element**
  - **ParametricElement**
    - **GeometricPrimitive**
      - **WireframePrimitive**
        - **Point**
        - **Line**
        - **Circle**
        - **Arc**
        - **ConicSection**
          - **Ellipse**
          - **Parabola**
          - **Hyperbola**
        - **NURBSCurve**
        - **Polygon**
      - **CSGPrimitive**
        - **Parallelepiped**
        - **Cylinder**
        - **Cone**
        - **Sphere**
        - **Torus**
        - **Polyhedron**
      - **BoundaryPrimitive**
        - **Plane**
        - **CylindricalSurface**
        - **ConicSurface**
        - **SphericalSurface**
        - **NURBSSurface**
    - **ConstructionElement**
      - **ConstructionPoint**
      - **ConstructionLine**
      - **ConstructionCircle**
      - **ConstructionPlane**
    - **Constraint**
      - **GeometricConstraint**
        - **StaticPoint**
        - **StaticCoordinate**
        - **EqualSizes**
        - **CoincidentPoints**
        - **CoincidentCoordinates**
        - **ParallelLines**
        - **PerpendicularLines**
        - **PlanarLines**
        - **ColinearLines**
        - **PointLiesOnLine**
        - **PointLiesOnCircle**
      - **DimensionalConstraint**
        - **Size**
        - **SizeProjection**
        - **AngleBetweenPoints**
        - **AngleBetweenLines**
        - **ProjectionOnLine**
        - **ProjectionOnCircle**
        - **Perimeter2D**
        - **Area2D**
        - **Centroid2D**
        - **Area3D**
        - **Volume3D**
        - **Centroid3D**
      - **AlgebraicConstraint**

Рис. 3. Иерархия классов элементов параметрического моделирования

Каждый из перечисленных классов наследует методы, декларируемые базовым классом ***Element***. Класс ***ParametricElement*** дополнительно определяет, а конкретные классы реализуют методы файлового ввода/вывода и визуализации параметрических элементов в окне для взаимодействия с базами данных и интерактивной графической библиотекой в рамках законченного приложения. В каждом классе ограничений переопределяется набор процедур, задающих математическую модель элемента (вычисление функциональных отображений и их производных). Для формирования модели всей системы используется класс ***StaticModel1*** инструментальной среды. Так как геометрические примитивы и конструкционные элементы не имеют собственных математических моделей, в соответствующих классах реализованы методы внесения дополнительных ограничений в параметризованную сцену.

Поскольку задача параметрического геометрического моделирования в большинстве случаев сводится к решению недоопределенных систем нелинейных алгебраических уравнений, возникает необходимость реализации специфического алгоритма для решения подобных систем. Мы используем нелинейный ABS-метод квазиньютоновского типа, основанный на модифицированном алгоритме Хуанга [18]. Для его реализации в объектно-ориентированную математическую библиотеку добавлен класс ***NonlinearABSAlgorithm***. При этом все остальные классы библиотеки остались без изменения.

Применение объектно-ориентированной инструментальной среды позволяет в максимальной степени унифицировать программные средства формирования и решения параметрических моделей. При этом обеспечиваются важные инструментальные возможности для их дальнейшего развития как в направлении обобщения постановок задач параметризации соответствующим расширением видов геометрических примитивов и ограничений, так и в направлении модернизации методов решения.

5.3. Реализация системы параметрического моделирования

Предложенная инструментальная среда апробирована при реализации графической интерактивной системы параметрического моделирования. Система реализована на языке Си++ в среде Win32/Win32s и включает в себя:

- объектно-ориентированную инструментальную среду для разработки систем численного моделирования,
- компоненты объектно-ориентированной математической библиотеки,
- библиотеку геометрических примитивов и ограничений,
- пользовательский многооконный интерфейс,
- графический редактор параметрических геометрических сцен.

Система позволяет параметризовать 2–D и 3–D геометрические модели с использованием базового набора примитивов и ограничений (см. рис. 4, 5). Геометрические сцены представлены каркасными моделями. Система допускает одновременное редактирование нескольких сцен и визуализацию одной сцены в нескольких стандартных видах (сверху, спереди, слева, изометрическом). Обеспечивается возможность сохранения и загрузки параметризованных геометрических сцен в обменных файлах во внутреннем формате. В настоящий момент в систему включены только численные методы моделирования, основанные на решении нелинейных систем алгебраических уравнений, и реализован режим счета по указанию пользователя.

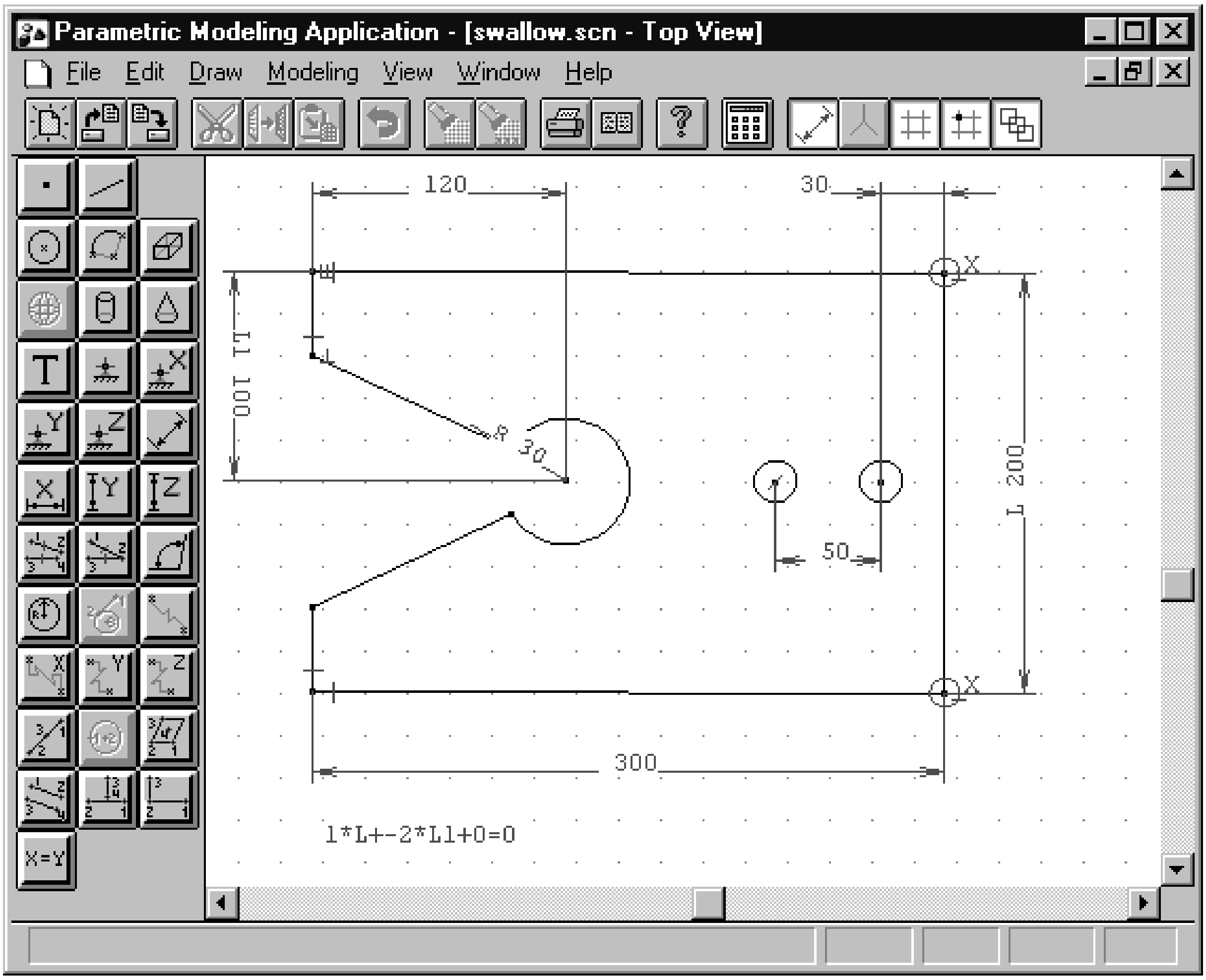

Рис. 4. Параметризованный плоский контур

Объектно-ориентированная архитектура является главной особенностью реализованной системы параметрического моделирования. Она позволяет легко наращивать возможности системы путем включения новых типов геометрических примитивов и ограничений, а также повышать эффективность и надежность моделирования путем включения новых алгоритмов. Более того, представляется возможным решать на той же самой методологической основе различные инженерные задачи, такие как эскизное проектирование, вариационная геометрия, моделирование сборок, моделирование физических свойств, оптимальное размещение компонентов конструкций, путем множественного использования программных средств параметризации.

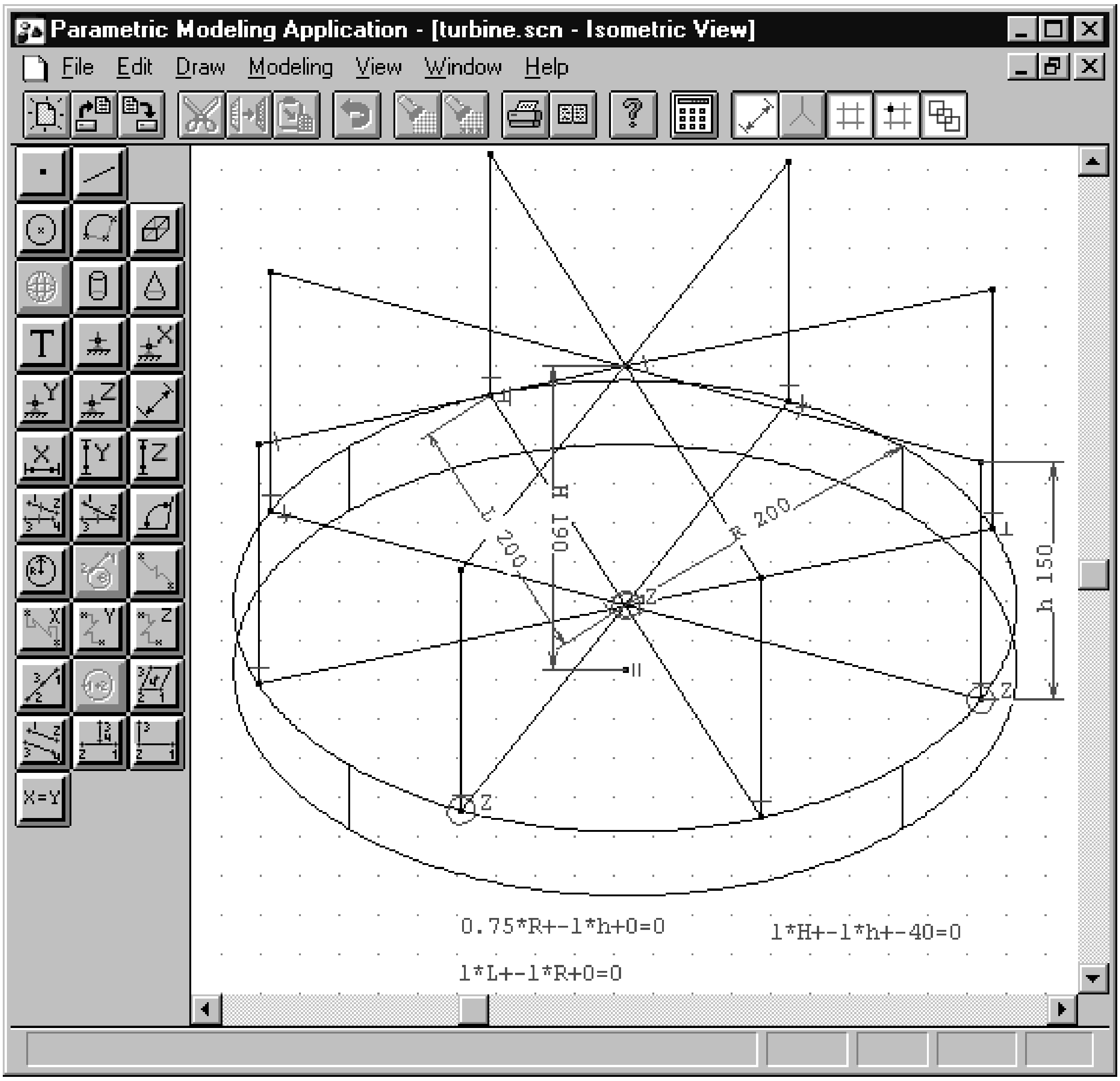

Рис. 5. Параметризованная каркасная модель турбины

6. Система моделирования течений в трубопроводных сетях

Разрабатываемая система моделирования течений в трубопроводных сетях предназначена для расчета различных инженерных систем, осуществляющих централизованное снабжение рассредоточенных потребителей электрической и тепловой энергией, топливом, водой или какой-нибудь другой транспортируемой средой.

Класс задач расчета таких сооружений чрезвычайно широк и включает в себя моделирование электроэнергетической и газоснабжающей систем страны; межрегиональных систем (объединенных энергосистем, магистральных нефте- и газопроводов, систем каналов и групповых водопроводов); систем электро-, тепло-, водо- и газоснабжения городов и промышленных центров; систем отопления, вентиляции и др.

6.1. Математическая постановка задачи и методы ее решения

Вопросы математической формализации решения задач проектирования трубопроводных и других гидравлических систем составляют предмет современной теории гидравлических цепей [20]. Отправной точкой теории служит тот факт, что данные объекты обладают топологической общностью своих расчетных схем, а движение транспортируемой среды в них подчиняется единым законам течения и сетевым законам сохранения массы и энергии.

Исходя из этой общности, гидравлической цепью принято считать совокупность устройств и соединяющих их трубопроводов, закрытых или открытых каналов, осуществляющих транспортировку сжимаемых и несжимаемых жидкостей (воды, нефти, газа, воздуха и других).

В любой гидравлической системе различают три основные группы компонентов: 1) источники давления или расхода (например, насосные или компрессорные станции, аккумулирующие емкости и др.), обеспечивающие притоки транспортируемой среды и привносящие энергию в систему; 2) трубопроводы и другие гидравлические элементы типа воздуховодов и открытых каналов, соединяющие источники с множеством потребителей и доставляющие эту среду; 3) потребители.

Для гидравлической цепи задаются ее технические характеристики (диаметры и длины трубопроводов, размеры сечений каналов, шероховатость), а также граничные условия — варьируемые входные величины притоков и нагрузок. Состояние гидравлической цепи характеризуется расходами среды в ее компонентах, а также давлением и температурой в узлах соединения.

Система моделирования течений должна позволять задавать различные технические характеристики в соответствии с ГОСТом для труб, насосов и т.п. Кроме этого, система должна предусматривать возможность использования различных математических моделей для каждого компонента. Например, для труб и насосов могут применяться как квадратичная, так и кубическая модели.

Математически состояние гидравлической цепи описывается функциональными характеристиками устройств, компонентов цепи и едиными законами сохранения. Данное обстоятельство позволяет нам рассматривать широкий класс гидравлических задач в рамках вышеизложенной концепции.

Вектор переменных состояний гидравлической цепи будет включать в себя давления узлов, являющиеся переменными связей, и потоки в компонентах — их переменные состояния. Формирование модели гидравлической цепи осуществляется на основе элементных уравнений и топологических уравнений Кирхгофа для суммарных потоков через соединения цепи. Данный способ формирования модели соответствует второму стандартному способу, который предусматривается концепцией.

6.2. Специализированная библиотека элементов

Для построения системы моделирования течений в сетях трубопроводов была использована описанная выше инструментальная среда моделирования. Разработка приложения свелась к реализации специальных классов элементов гидравлической цепи. Библиотека классов организована в виде иерархии классов, наследуемых от базового класса ***Element*** (см. рис. 6).

- **Element** (абстрактный элемент)
  - **PipingElement** (элемент гидравлической цепи)
    - **Pipe** (труба)
    - **HeatPlant** (ТЭЦ)
    - **Tributary** (приток жидкости)
    - **Consumer** (потребитель)
    - **Pressure** (источник давления)
    - **Pump** (насос)
      - **Pump1**
      - **Pump2**
      - **Pump3**
    - **Reservoir** (резервуар)
    - **Damper** (гаситель энергии потока жидкости)
    - **Regulator** (регулятор)
      - **PressureRegulator** (регулятор давления)
      - **FlowRegulator** (регулятор расхода)
    - **Torrent** (поток жидкости)
    - **Radiator** (радиатор отопления)

Рис. 6. Иерархия классов элементов гидравлических цепей

Каждый из классов компонентов может служить родительским (базовым) классом для однотипных элементов, имеющих общие характеристики и параметры. Например, класс ***Pump*** является родительским классом для трех различных моделей насосов.

Для формирования статической модели гидравлической системы используется стандартный класс ***StaticModel2***, входящий в состав среды.

6.3. Реализация системы

Система моделирования течений в трубопроводных сетях разработана на языке Си++ в среде Win32/Win32s. Система состоит из следующих компонентов:

- объектно-ориентированной среды моделирования,
- специализированной библиотеки элементов,
- графического интерфейса пользователя,
- графического редактора гидравлических цепей.

Для создания оконного интерфейса использовались разработанные классы общей системы, а также были определены методы символического отображения компонентов гидравлической цепи. Для визуализации результатов расчета гидравлической цепи использовались специализированные методы. Для выбранных пользователем элементов цепи отображаются значения потоков через эти элементы, а для выбранных узлов — значения давлений в них (см. рис. 7). Возможности редактирования, такие как выделение, удаление элементов, поддерживаются самой инструментальной средой.

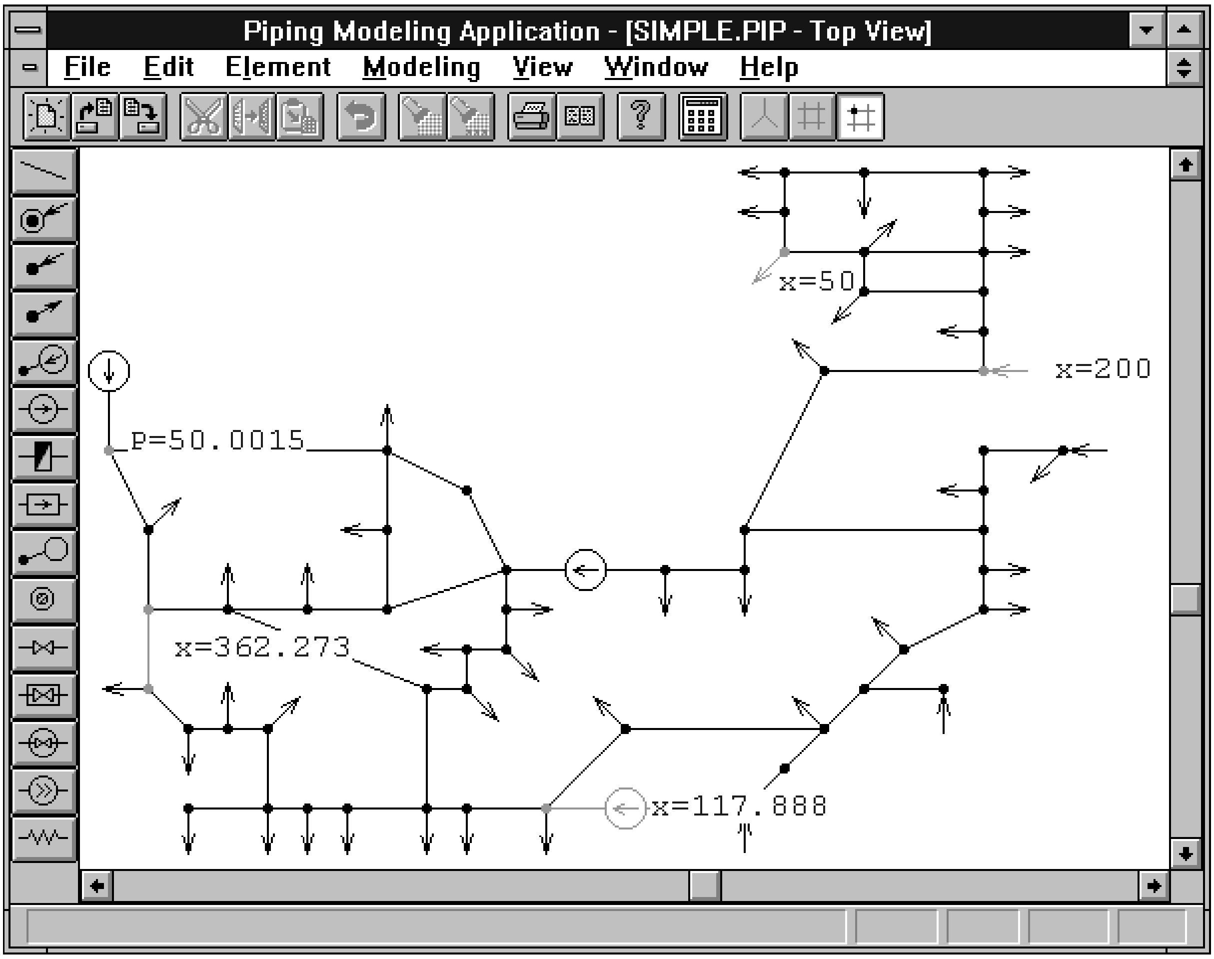

Рис. 7. Пример модели трубопроводной сети

7. Заключение

В настоящее время реализованная на языке Си++ объектно-ориентированная инструментальная среда обеспечивает разработку средств моделирования нелинейных статических систем. На основе данной среды были спроектированы и реализованы демонстрационные версии прикладных систем:

- параметрического геометрического моделирования,
- моделирования течений в трубопроводных сетях.

Практический интерес представляет также моделирование нелинейных динамических стохастических систем. В связи с этим являются важными обобщение предложенной концепции моделирования на случай подобных систем и полная реализация инструментальной среды моделирования. Не менее актуальными являются поиск и параллельная разработка новых программных приложений с использованием данной среды.

В большинстве случаев усилия на адаптацию среды моделирования к конкретной предметной области относительно малы. Для сложных, но распространенных приложений разработка специализированных классов может осуществляться централизованно, что также снизит затраты на создание законченных программных систем.

Таким образом, предложенный подход к численному моделированию представляется перспективным для эволюционной разработки CAD/CAM систем в различных предметных областях.

Работа поддержана Российским Фондом фундаментальных исследований (грант 95–01–01239).

ЛИТЕРАТУРА

1. Proceedings of the Second Annual Object-Oriented Numerics Conference, Apr. 24–27, 1994, Sunriver, OR.
2. Proceedings of the Conference on Parallel Object-Oriented Methods and Applications, Feb. 28 – Mar. 1, 1996, Santa Fe, NM.
3. Proceedings of the 1996 ACM SIGPLAN Conference on Object-Oriented Programming: Systems, Languages, and Applications, Oct. 6–10, 1996, San Jose, CA.
4. Крон Г. Исследование сложных систем по частям — диакоптика. — М.: Наука, 1972.
5. Хэпп Х. Диакоптика и электрические цепи: Пер. с англ. — М.: Мир, 1974.
6. Баталов Б.В., Егоров Ю.Б., Русаков С.Г. Основы математического моделирования больших интегральных схем на ЭВМ. — М.: Радио и связь, 1982.
7. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными: Пер. с англ. — М.: Мир, 1975.
8. Акритас А. Основы компьютерной алгебры с приложениями: Пер. с англ. — М.: Мир, 1994.

9. Семенов В.А. Об объектно-ориентированном подходе к разработке численного математического обеспечения. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1995, с. 140–163.
10. Семенов В.А., Морозов С.В. Объектно-ориентированное программирование задач численного анализа. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1995, с. 189–211.
11. Семенов В.А., Тарлапан О.А. Технологии реализации разреженных матричных классов. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1995, с. 164–188.
12. Семенов В.А., Ширяева Е.Ю. Объектная классификация задач и методов нелинейной безусловной оптимизации. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1996, с. 86–119.
13. Семенов В.А., Морозов С.В. Объектно-ориентированное программирование квадратурных методов. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1996, с. 120–146.
14. Семенов В.А., Тарлапан О.А. Объектно-ориентированный подход к программированию прямых методов линейной алгебры. // Вопросы кибернетики. Приложения системного программирования. — М.: НСК РАН, 1996, с. 147–170.
15. CADDS 5 Cvware Parametric Design. Technical Summary, Revision 1.0. Computervision Corporation, Bedford, MA, 1991.
16. Light R., Gossard D. Modification of geometric models through variational geometry. // CAD, v. 14, № 4 (July 1982), pp. 209–214.
17. Freeman-Benson B.N., Maloney J., Borning A. An incremental constraint solver. // Comm. of the ACM, v. 33, № 1 (Jan. 1990), pp. 54–63.
18. Абаффи Й., Спедикато Э. Математические методы для линейных и нелинейных уравнений: Проекционные ABS-алгоритмы: Пер. с англ. — М.: Мир, 1996.
19. Чуа Л.О., Пен-Мин Лин. Машинный анализ электронных схем: Пер. с англ. — М.: Энергия, 1980.
20. Меренков А.П., Хасилев В.Я. Теория гидравлических цепей. — М.: Наука, 1985.