

УДК 303.732

ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ВИЗУАЛЬНОЙ РАЗРАБОТКИ ИМИТАЦИОННЫХ МОДЕЛЕЙ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

(Работа выполнена при финансовой поддержке РФФИ, проекты 15-08-01886-а, 15-01-03067-а)

М.А. Бутакова, д.т.н., профессор, butakova@rgups.ru; А.Н. Гуда, д.т.н., профессор, проректор по научной работе, guda@rgups.ru; А.В. Чернов, д.т.н., профессор, зав. кафедрой, avche@yandex.ru

(Ростовский государственный университет путей сообщения, пл. им. Ростовского Стрелкового полка Народного ополчения, 2, г. Ростов-на-Дону, 344038, Россия)

Аннотация. Статья посвящена результатам разработки проблемно-ориентированной визуальной системы моделирования сетевых информационных систем на железнодорожном транспорте. Подробно рассмотрены наиболее распространенные программные средства имитационного моделирования, основное внимание уделено системам, позволяющим моделировать сети массового обслуживания. Описаны теоретические аспекты моделирования сетей массового обслуживания, языки и средства, позволяющие реализовать эти модели в программных средствах. Рассмотрен проблемно-ориентированный программный комплекс, предназначенный для моделирования сетевых информационных систем на железнодорожном транспорте средствами имитационных моделей сетей массового обслуживания.

Ключевые слова: *имитационное моделирование, проблемно-ориентированное моделирование, визуальное моделирование, сеть массового обслуживания, информационная система.*

Уже более полувека общепризнанным инструментом научных исследований является *имитационное моделирование* (ИМ) с использованием средств компьютерной техники. Все эти годы методы и алгоритмы компьютерного ИМ обсуждались в десятках тысяч научных работ, и до настоящего времени исследования в этом направлении продолжают весьма интенсивно. Разные категории научных работников, особенно молодые ученые, хорошо осведомленные в информационных технологиях, предлагают различные компьютерные алгоритмы и программы для расчета параметров и режимов еще не созданных систем в промышленности, на транспорте и в других областях человеческой деятельности. Имитационные модели как частный случай математических моделей могут относиться к различным областям математики, но среди прочих есть общепризнанные, в число которых входят модели *сетей массового обслуживания* (СеМО). Эти модели хорошо подходят для задач моделирования сетевых *информационных систем* (ИС), в том числе на транспорте [1].

В данной работе ставится цель разработки единой среды визуального описания моделей и интегрированной моделирующей системы с предметно-ориентированной областью сетевых ИС на железнодорожном транспорте. Предыдущие исследования показывают, что адекватными моделями информационных потоков в ИС на железнодорожном транспорте являются модели случайных процессов специального вида, такие как случайные процессы типа Леви [2], автомодельные случайные процессы [3]. Эти процессы генерируют входные потоки для СеМО, моделирующей функционирование сетевых ИС. Целесообразность разработки нового подхода в сравнении с существующими системами и языками ИМ обусловлена несколькими причинами. Во-первых, известные системы ИМ требуют от исследователя высокой математической подготовки даже на предварительных этапах описания моделируемых задач. Вторая причина – необходимость простоты перенастройки параметров модели, которую желательно воплощать визуально. Третья причина, скорее, связана с проблематикой данной работы и состоит в том, что в существующих системах ИМ отсутствуют средства анализа и оценки систем с помощью аппарата СеМО, к тому же нет возможности анализа модели под воздействием случайных процессов (а не вероятностных распределений). Четвертая причина – повышение адекватности моделирования процессов, происходящих в различных ИС, у которых базовая часть ПО и операционная система функционируют по принципам объектно-ориентированного программирования. Известные системы ИМ представляют модель параметрически-функционально, а не объектно-ориентированно, таким образом несколько отставая от реальной проблематики моделирования. Последний фактор связан с требованиями повышения эффективности разработки имитационных моделей с возможностями представления их в виде СеМО, а также модификации структуры программы без изменения ее функциональности.

Имитационное и визуальное моделирование: подходы и методологии, стандарты и спецификации

В настоящее время имеются значительный опыт и многочисленное ПО, позволяющее проводить компьютерные эксперименты в области моделирования технических систем и технологических процессов.

Разработаны ПО, языки описания и графического моделирования структуры комплексов программ, предназначенные для автоматической генерации компонентов ПО. Тем не менее, в области компьютерного моделирования существуют устоявшиеся принципы, стандарты и спецификации моделирования, среди которых следует выделить наиболее распространенные и значительно превосходящие остальные по своим возможностям системы.

За всю историю развития вычислительной техники было создано более 300 языков моделирования дискретных процессов. Одним из первых языков, появившихся в начале 1960-х годов, был язык блок-диаграмм, предложенный Дж. Гордоном [4, 5]. Его идеи оказались настолько плодотворными, что использовались в многочисленных разработках в нашей стране и за рубежом. На основе этого языка в 1970-х годах были созданы и в последующем адаптированы к персональному компьютеру широко используемый в настоящее время для моделирования большого класса систем язык и система моделирования GPSS (*General Purpose Simulation System*) [6].

Модели, разрабатываемые на GPSS, компактны и часто состоят из меньшего числа операторов, чем аналогичные модели, написанные на процедурных языках программирования. Это объясняется тем, что в него встроено большое число логических подпрограмм, используемых для моделируемых систем. В GPSS также входят специальные средства для описания динамического поведения систем, причем изменение состояний происходит в дискретные моменты времени. Многие функции этой системы моделирования выполняются автоматически, например, сбор описывающих поведение модели итоговых статистических данных и их выдача по завершении процесса моделирования. Встроенный язык моделирования, называемый GPSS/PC, построен в предположении, что моделью сложной дискретной системы является описание ее элементов и логических правил их взаимодействия в процессе функционирования моделируемой системы. Для определенного класса систем можно выделить небольшой набор абстрактных элементов языка – объектов. Описание модели на GPSS представляет собой последовательность блоков, каждый из которых имеет определенное количество реквизитов, называемых полями. Порядок следования блоков в программе строго определен, а положение полей строго фиксировано. Операторы языка GPSS/PC относятся к одному из четырех типов: карты описания блоков (формируют логику модели), карты описания объектов (описывают параметры объектов), управляющие карты (управляют процессом моделирования), управляющие карты системы (управляют интегрированной средой GPSS). Готовая модель транслируется в текущую модель, являющуюся совокупностью объектов разного типа. Каждому объекту соответствуют числовые атрибуты, описывающие его состояние в данный момент модельного времени. Часть атрибутов доступна программисту, они называются стандартными числовыми атрибутами. Значения атрибутов могут быть арифметическими или логическими. Имеется также ряд системных атрибутов, относящихся не к отдельным объектам, а к модели в целом.

Имитация модели происходит следующим образом. Конфигурация блок-схемы GPSS-модели отражает направления, по которым перемещаются элементы, например заявок в СеМО. Каждый такой элемент называется транзактом. В начальный момент времени нет ни одного транзакта, а в процессе моделирования они поступают в соответствии с логическими потребностями, которые возникают в моделируемой системе. Следует заметить, что при этом на исследователя возлагается задача адекватного сопоставления потока транзактов в модели с некоторым вероятностным распределением. В общем случае в модели существует большое число транзактов, но в некоторый момент времени перемещается только один из них. В процессе продвижения транзакта возможно одно из следующих событий: транзакт входит в блок, функцией которого является удаление его из модели; транзакт входит в блок, функцией которого является задержка его на некоторое время; транзакт пытается войти в блок в соответствии с блок-схемой, но блок отказывается его принять. По сути моделирующая система работает с двумя списками событий: текущих, у которых время продвижения транзакта меньше или равно модельному, и будущих, для которых время продвижения больше модельного. В зависимости от условий в модели транзакты переносятся из списка в список и корректируется таймер модельного времени. Реализацию моделирования в GPSS можно охарактеризовать следующим образом. По типу моделирования эта система относится к событийному моделированию. По организации модельного времени – выполняется имитация последовательностей переходов в целочисленные интервалы времени. Приращение модельного времени на каждом шаге моделирования – величина случайная, равная интервалу времени между двумя последовательными событиями. По типу имитации процессов, протекающих в моделируемой системе, GPSS представляет собой систему управления списками событий. Алгоритм событийного моделирования состоит в формировании, просмотре и изменении этих списков.

В контексте данного исследования использование исключительно системы GPSS невозможно по ряду причин. Во-первых, традиционными недостатками этой системы являются сложность освоения и отсутствие наглядности языка описания моделей для его применения инженерно-техническими работниками. Во-вторых, закрытость системы не позволяет добавить новые виды распределений потоков заявок. Третий фактор – невозможность задания процесса, происходящего в моделируемой системе, как стохастического (например, процесса типа Леви или другого). Кроме того, являясь одной из первых классических

систем моделирования, современные версии GPSS до сих пор поддерживают процедурно-событийные, а не объектно-ориентированные методы моделирования. Тем не менее, некоторые базовые принципы, составляющие функциональность этой системы, в усовершенствованном виде использованы при создании среды быстрой разработки и моделирования информационных систем, имитационного и математического моделирования с использованием визуального и объектно-ориентированного проектирования.

В конце 1980-х годов широкое распространение получила система MATLAB (*MATrix LABoratory*) – интерактивная система, предназначенная для выполнения инженерных и научных расчетов и ориентированная на работу с массивами данных [7]. В настоящее время эта система принята в качестве официального средства оформления инженерной документации и научных публикаций. MATLAB специально создана для проведения именно инженерных расчетов. Используемый в ней математический аппарат максимально приближен к современному математическому аппарату инженера и ученого и опирается на вычисления, производимые с матрицами, векторами и комплексными числами. Графическое представление функциональных зависимостей организовано в ней в форме, отвечающей инженерной документации. Язык программирования MATLAB достаточно прост, он содержит всего несколько десятков операторов, что компенсируется большим числом процедур и функций, содержание которых понятно пользователю. Важным отличием от других математических пакетов программ является открытость системы: практически все процедуры и функции доступны не только для использования, но и для модификации. Основной объект системы MATLAB – прямоугольный числовой массив (матрица), в котором допускается применение комплексных элементов. MATLAB предоставляет широкие возможности для работы с сигналами, включая построение их частотных, импульсных и переходных характеристик. В системе есть также средства выполнения спектрального анализа и синтеза, включая реализации прямого и обратного преобразований Фурье. Система MATLAB использует собственный М-язык, сочетающий в себе положительные свойства различных известных языков программирования высокого уровня. От языка Pascal система MATLAB позаимствовала объектно-ориентированную направленность, то есть такое построение языка, которое обеспечивает образование новых типов вычислительных объектов на основе уже существующих (принцип наследования). Новые типы объектов в MATLAB, как и в современных системах программирования, называются *классами* и могут иметь собственные процедуры их преобразования (определяют методы этого класса). MATLAB обладает возможностью, отсутствующей в других языках: ее интерпретатор в одном и том же сеансе работы позволяет выполнять несколько самостоятельных программ, причем все переменные, используемые в этих программах, являются для них общими и образуют единое рабочее пространство.

Наиболее интересной и полезной особенностью MATLAB в контексте данного исследования является наличие в ней наглядного и эффективного средства разработки программных моделей – пакета визуального программирования Simulink. Этот пакет позволяет осуществлять исследование (моделирование во времени) поведения динамических линейных и нелинейных систем. Разработку программы и ввод характеристик систем можно производить в диалоговом режиме путем сборки на экране схемы соединений элементарных (стандартных или пользовательских) звеньев. В результате такой сборки получается модель системы (S-модель), которая сохраняется в файле. Создание моделей в пакете Simulink основывается на технологии Drag-and-Drop. В качестве элементарных звеньев при построении S-модели используются визуальные блоки (модули), сохраняющиеся в специальной библиотеке. Эта библиотека может быть дополнена пользователем за счет разработки собственных блоков. S-модель может иметь иерархическую структуру, то есть состоять из моделей более низкого уровня, причем количество уровней иерархии практически не ограничено. При моделировании есть возможность наблюдать за процессами, происходящими в системе. Для этого используются специальные блоки («обзорные окна», Data Viewers), входящие в состав пакета Simulink. Однако следует заметить, что визуальные блоки этого пакета нельзя отождествлять с визуальным компонентным подходом к разработке и моделированию программных комплексов, так как ими не поддерживается объектно-ориентированная технология. Этот факт затрудняет применение пакета Simulink для моделирования рассматриваемых в данной работе систем в условиях изменения множества внешних факторов. Блоки Simulink, из которых составляется модель, обладают некоторым конечным набором параметров и детерминированными физическими и математическими законами, описывающими поведение блока, что не подходит для моделирования стохастических систем, каковыми являются практически все телекоммуникационные и информационные системы.

Объектно-ориентированное визуальное моделирование как одна из областей компьютерной инженерии стало активно развиваться в начале 90-х годов. Наибольшее влияние на формирование этой области оказали исследования Г. Буча, И. Джакобсона, Д. Рэмбо, П. Коуда, Д. Харела, Б. Селика и др. Их усилиями был создан стандарт в этой отрасли – язык UML (*Unified Modeling Language*) [8]. С тех пор UML был принят и поддержан большинством производителей компьютерных средств моделирования, а после принятия организацией Object Management Group в 1996 г. он стал утвержденным промышленным стандартом. UML также помог сформировать всеобщее понимание значения моделирования при работе со сложным ПО. Этот язык моделирования сужает разрыв между ПО и системами, которые оно моделиру-

ет. Целью применения UML является формализованная связь модели с ее соответствующей программной реализацией через одно или несколько автоматизированных преобразований модели. Комбинация абстракций и автоматизации привела к появлению новых технологий компьютерного моделирования и соответствующих методов разработки комплексов программ, относящихся к методам управляемой модели разработки (*Model-Driven Development*). В этой методике модели становятся основными объектами дизайна комплекса программ, перемещая, таким образом, внимание разработчика от кода программы на семантическое содержание модели. Степень автоматизации, используемая в указанном методе, варьируется от простого скелетного кода до генерирования полного автоматического кода комплексов программ. Семантическая среда UML версии 2.0 включает следующие элементы. *Метамодель* определяется набором формальных ограничений, написанных на языке Object Constraint Language, и изображается в виде *диаграмм классов* Meta-Object Facility. *Активности* используются в UML для моделирования потоков различного типа: потоков сигналов или данных, а также алгоритмических или процедурных потоков. *Взаимодействия* в UML предназначены для моделирования последовательностей действий произвольной сложности. С помощью *конечных автоматов* выполняется моделирование управляемого событиями поведения системы.

Язык UML реализуется в программных средствах, называемых CASE-системами. Термин CASE трактуется весьма широко. Первоначально он расшифровывался как *Computer Edit Software Engineering* (компьютерная поддержка разработки ПО), и связано это было с автоматизацией разработки программных систем. В дальнейшем понятие CASE приобрело новый смысл. Все чаще его стали расшифровывать как *Computer Edit System Engineering*, а CASE-средства все больше стали ориентироваться на создание спецификаций проектирования и моделирования сложных систем широкого назначения. Современные CASE-средства охватывают обширную область поддержки многочисленных технологий проектирования программных систем от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл ПО. Наиболее трудоемкими этапами разработки являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение структурных или иных диаграмм в реальном масштабе времени, сквозную проверку синтаксических правил. Графические средства моделирования предметной области позволяют разработчикам в наглядном виде изучать существующую ИС, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями. Следует отметить, что среди отечественных программных разработок направление CASE-средств не поддерживается, а зарубежные, такие как IBM Rational Rose, требуют от пользователей специальной подготовки и обучения. Учитывая данное обстоятельство, разработанный в данной главе программный комплекс использует CASE-методологию, но в доступной для пользователя форме.

Применительно к задачам формализации проектирования и разработки телекоммуникационных систем был создан язык SDL (*Specification and Description Language*) – язык спецификаций и описаний, предназначенный для разработки событийно-ориентированных распределенных систем. Под спецификацией понимается точное формальное определение системы или ее части, под описанием – неформальная спецификация, иллюстрирующая тот или иной аспект системы. Описания используются на ранних этапах разработки системы или для ее документирования, спецификации – на стадии детального проектирования, и по ним предполагается автоматическая генерация программного кода. Тот факт, что для этих разных этапов разработки системы предлагается один язык, является несомненным достоинством SDL, поскольку в этом случае преодолевается проблема семантических разрывов. SDL развивается международным комитетом ITU с 1976 года. Есть два варианта этого языка – текстовый (SDL/PR) и графический (SDL/GR), семантика которых, за исключением некоторых тонкостей, совпадает. Основным документом по SDL является [9]. Изложение основ этого языка можно найти в русскоязычных работах [10, 11].

Кроме языка SDL, комитет ITU предложил целое семейство стандартов на средства разработки телекоммуникационных систем. Можно назвать язык высокого уровня CHILL [12], MSC [13] (графический язык сценариев). В работе [14] содержатся рекомендации по согласованному использованию упоминаемых выше стандартов.

Язык SDL как средство анализа систем широко используется в европейских телекоммуникационных стандартах. Его основными составляющими являются структурная модель и расширенный конечный автомат. Они ориентированы на спецификацию событийно-ориентированных систем, допускается, впрочем, и более широкое их использование. В основе структурной декомпозиции системы при помощи SDL лежит блочный анализ. Его результатом является разбиение системы на вложенные друг в друга части (блоки), которые не содержат исполняемый код, а только одни описания. Они могут соответствовать крупным модулям системы или подзадачам проекта. Исполняемый код в виде расширенного конечного автомата содержится лишь в листьях этой декомпозиции – процессах, которые, как и блоки, можно сопоставить объектам. Поэтому SDL был успешно расширен до объектно-ориентированного языка. В аспекте данного исследования заметим, что на сегодняшний день в языке SDL не предусмотрены разработка и

спецификация моделей для телекоммуникационных систем в условиях стохастических процессов, в частности, фрактального трафика.

В заключении сделанного обзора укажем, чем обосновывается целесообразность практической реализации среды быстрой разработки и моделирования информационных систем, имитационного и математического моделирования с использованием визуального и объектно-ориентированного проектирования. Первый аспект заключается в том, что необходимо выполнить проверку адекватности предложенных в предыдущих главах математических моделей, методов и алгоритмов, а также стохастического подхода к моделированию ИС в целом. Второй аспект заключается в сочетании возможностей системы ИМ с методами графического визуального моделирования и объектно-ориентированного компонентного подхода к разработке программных комплексов. Третий аспект – очевидная необходимость повышения эффективности разработки имитационных моделей с возможностями модификации и управления процессом анализа и проектирования ИС.

Общие положения ИМ

Область ИМ является достаточно обширной и проработанной на протяжении нескольких последних десятилетий, за которые накоплен позитивный опыт. Рассмотрим подробнее существенные идеи ИМ как базу для построения новых концепций моделирования информационных процессов. СеМО оказываются подходящим инструментом для ИМ по нескольким причинам. Во-первых, с помощью элементов графических обозначений СеМО можно визуальным образом отобразить структуру изучаемой системы, что придает имитационной модели свойство наглядности. Во-вторых, СеМО позволяет произвести иерархическую декомпозицию моделируемой системы, причем составные блоки будут унифицированными (то есть содержать определенное число параметров) на всех уровнях декомпозиции. Это свойство позволяет также осуществлять обмен данными между блоками, моделируемыми как отдельные СеМО. В-третьих, расчет параметров узлов СеМО «мгновенно», в реальном времени, а точнее, в модельном времени, позволяет имитировать процессы, происходящие в моделируемой системе при наличии любых произвольных законов распределения входных параметров моделей узлов СеМО.

Рассмотрим конкретные параметры узлов СеМО при их формализации в виде имитационных моделей. Входными параметрами для имитационного блока должны быть, по крайней мере, следующие:

- промежуток времени, в течение которого проводится наблюдение (системное время);
- таймер модельного времени, позволяющий выполнять имитацию модели либо пошагово, либо блоком итераций;
- количество заявок (прибытий, поступлений, запросов в зависимости от физической природы моделируемого объекта), наблюдаемых на входе узла СеМО;
- количество завершаемых узлом действий, выполняемых по отношению к поступающим заявкам, операций обслуживания; причем действия могут быть направлены как на обслуживание, так и на игнорирование или частичное обслуживание заявок.

Перечисленные характеристики, за исключением системного времени, которое логично принимать за основу модельного времени имитационной системы, могут задаваться произвольно, а точнее – в соответствии с любым, желаемым для исследователя законом распределения или стохастическим процессом.

На основе этих характеристик и в соответствии с базовыми положениями теории систем и сетей массового обслуживания можно в первом приближении определять такие характеристики и их размерности, как *интенсивность поступления заявок* – число заявок в единицу времени, а также *производительность узла СеМО* – количество операций обслуживания в единицу времени.

Если узел СеМО имеет один обслуживающий прибор, то он может либо выполнить обслуживание (быть занятым), либо простаивать (быть свободным), тогда исследователь может измерить эту величину (при имитационном моделировании она является случайной) и найти *интервал времени занятости узла обслуживания*. Далее можно рассчитать *коэффициент использования узла СеМО* (разделив последний показатель на единицы системного времени). Также можно определить такую характеристику, как *среднее время, требуемое для обслуживания одной заявки прибором узла СеМО*. При моделировании системы в целом требуется «расширить» имитационную модель от единичных заявок на классы и потоки заявок. Например, среди заявок, обслуживаемых вычислительной системой, моделируемой как один узел СеМО, могут быть заявки, относящиеся к классу операций с жестким диском, с выводом на монитор, обращениями к компьютерной сети. Таким образом, при моделировании часто используется допущение о пропорциональности, соблюдении баланса между величинами прибытий заявок в узел СеМО и выполненных действий по их обслуживанию. Заметим, что здесь речь идет не о строгом равенстве, так как в реальных моделируемых системах возможны потери и игнорирование обслуживания заявок при некоторых условиях, а о том, что за некоторое модельное время поступающий поток заявок будет в целом обслуживаться узлом СеМО и передаваться далее, а не игнорироваться или задерживаться.

На основании приведенных рассуждений предполагается, что имитационную модель СеМО можно представлять в виде набора типовых элементов – узлов СеМО с их предопределенными характеристиками.

Визуальная среда моделирования ИС на транспорте

Разработанная авторами визуальная среда моделирования позволяет исследовать систему с помощью представления модели в трехуровневом виде:

- 1) уровень формализации описания исследуемой системы и процессов;
- 2) уровень идентификации математической модели со структурными элементами и потоками данных в системе;
- 3) уровень рефакторинга, позволяющего выполнять модификацию структуры программы без изменения ее функциональности, введение новых требований в систему, которые влекут изменение ее структуры, логической организации кода на уровне классов и кода внутри класса.

При моделировании ИС, описываемой в виде СеМО, будем представлять ее как совокупность элементов некоторого типового набора узлов, называемых *объектами*. Объектный принцип обеспечивает единообразие моделирования и свойства конструктивизма и модифицируемости. В разработанной среде реализуется объектный принцип, заключающийся в следующем. Все объекты изменяются на интервале времени Δt независимо друг от друга, который в дальнейшем будем называть *тактом моделирования*. По истечении такта моделирования объекты обмениваются информацией о своих состояниях, которые являются входной информацией для следующего такта моделирования. После выполнения одного такта можно произвести еще один, и так далее, то есть объекты рассчитываются в цикле. Главное, что отличает моделирующую объектную среду от объектной парадигмы программирования, это то, что каждое свойство объекта вычисляется на каждом шаге, то есть существует всегда, а абстрактных объектов и их характеристик нет. Чтобы объекты влияли друг на друга, необходимо проимитировать их связи. Для этого необходимо указать, каким образом связаны переменные величины характеристик различных объектов. Каждая переменная должна иметь два состояния: на начало такта и на его окончание, выполняя двухфазную схему моделирования. При этом причинно-следственная временная связь разделена во времени на один такт. Чаще всего принимается, что размер такта неизменный и, чем меньше шаг, тем больше точность, а чем больше шаг, тем быстрее идет процесс моделирования.

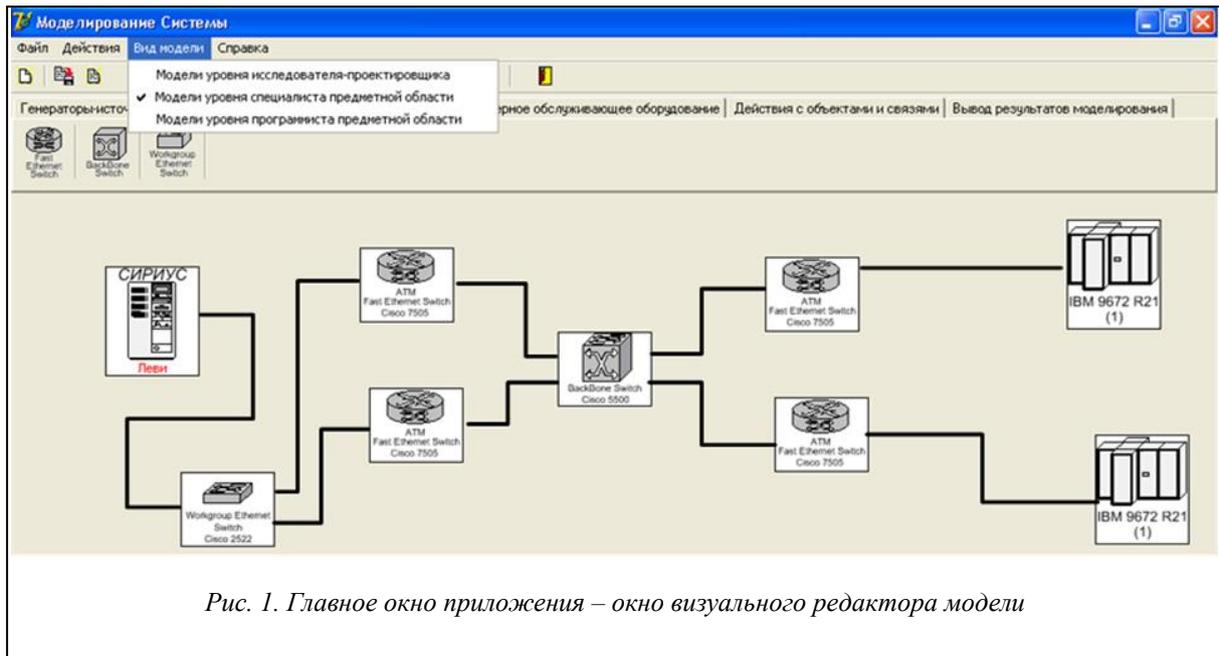
В разработанной среде представлены несколько видов информации: условно-графическая, математическая, матричная, ссылочная. Все виды информации доступны из модели и управляются ею через переменные или функции. Моделирование начинается с проектирования информационной схемы. Далее отдельные элементы подвергаются настройке и конкретизации. Затем выбирается способ взаимодействия элементов модели. Визуализация в разработанной в данном исследовании среде ведет к возможности управления моделью, принципиально упрощаются процессы настройки модели пользователями. Очевидно, что за визуальными объектами стоят математические модели, о которых шла речь выше. Визуализация позволяет быстро воспользоваться встроенным инструментарием для отображения свойств модели, а стандартные изображения в виде рисунка фиксированного размера можно поменять. В системе реализована математическая графика в виде гистограмм, двумерных графиков отображения случайных процессов и статистических оценок. Доступны все необходимые операции с визуальными объектами: создание, модификация, перемещение, удаление. Визуальная методика отображения модели принципиально упрощает и ускоряет проведение экспериментов с моделью.

В разработанной визуальной среде имитационного моделирования реализованы все этапы проведения компьютерных экспериментов:

- выделены и описаны наиболее существенные объекты, необходимые для построения модели ИС;
- определены законы протекания информационных процессов и условия их использования;
- реализованы правила оценки статистических характеристик параметров модели;
- разработаны схемы моделей, реализован этап установления иерархии и связей между объектами модели;
- возможна настройка численных значений параметров модели вместе с их корректировкой;
- реализован механизм проведения имитационного эксперимента с моделью.
- выполнено сопоставление абстрактных и математических показателей объектов модели с существующими в предметной области реальными техническими системами и устройствами.

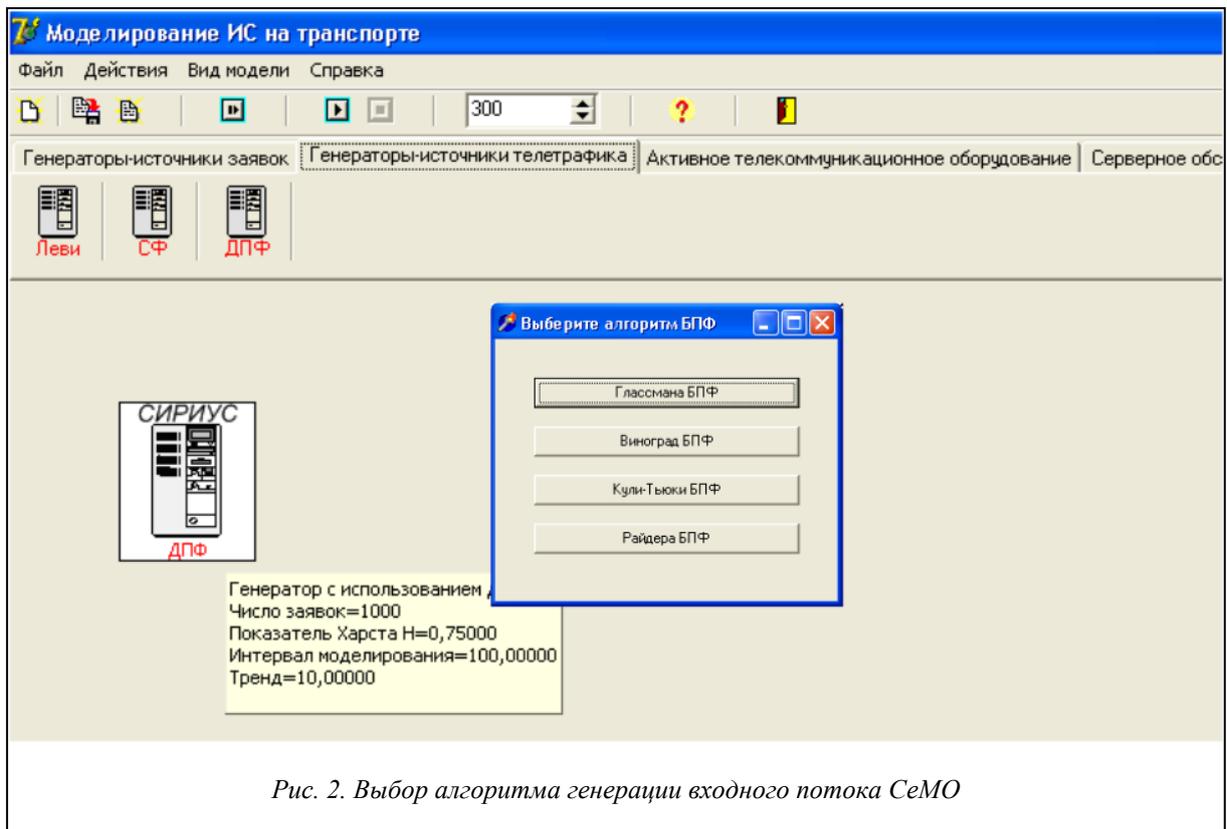
Подготовка описаний моделируемых процессов, задание режимов и параметров моделирования и вывода результатов ведутся с помощью встроенного редактора (см. рис. 1).

Для удобства работы с моделью создается новый проект – логическое объединение информационных материалов и ресурсов проекта, к которым относятся графическое изображение структурных элементов модели, библиотеки классов и объектов, текущие настройки среды и состояние переменных. Объектно-ориентированный подход разработки в среде подразумевает наличие базовых элементов модели в виде



объектов и классов, которые визуально отображаются с помощью определенного графического компонента. В разработанном в ходе данного исследования программном комплексе таковыми являются следующие:

– генераторы источников заявок в виде вероятностных распределений и случайных процессов – модели трафика в ИС (см. рис. 2);



– узловые устройства переключения маршрутов следования заявок – модели активного телекоммуникационного оборудования (см. рис. 3);

– устройства накопления и обслуживания заявок – модели серверов ИС (см. рис. 4).

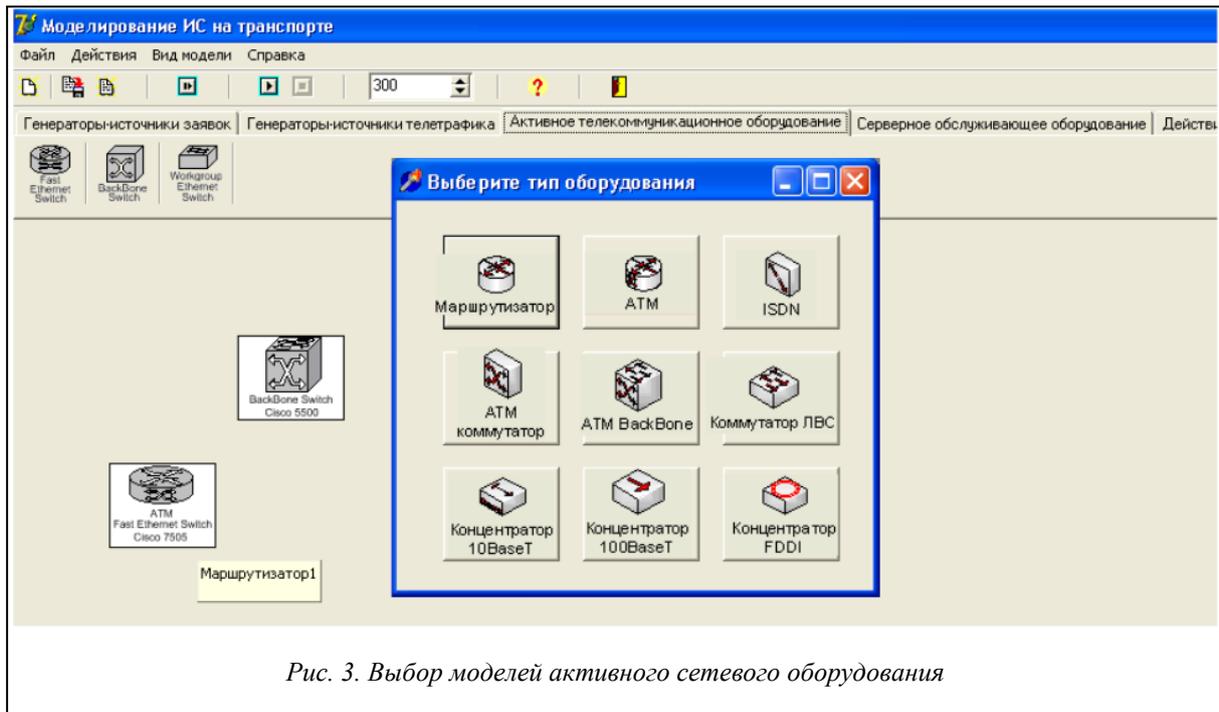


Рис. 3. Выбор моделей активного сетевого оборудования

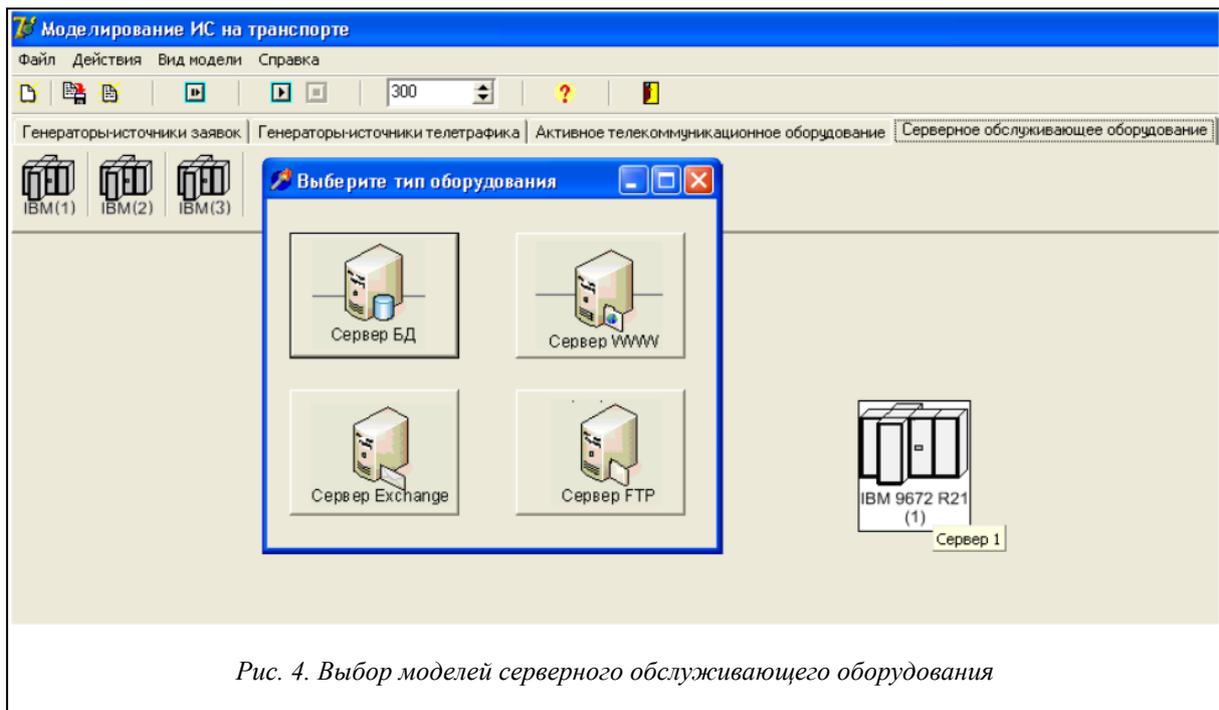


Рис. 4. Выбор моделей серверного обслуживающего оборудования

Принцип составления модели заключается в визуальном отображении с помощью блоков указанных классов структуры ИС заданной конфигурации и соединении их между собой для обозначения маршрута передачи информационных потоков. При построении модели создаются список классов и дерево объектов начиная от корневого объекта и соблюдается принцип наследования свойств и методов. Корневой объект стоит на вершине иерархии объектов проекта, а с точки зрения пользователя, он является прототипом моделируемой системы. Все остальные объекты просто встраиваются в схему корневого и так далее, образуя дерево иерархии объектов. Авторы используют подход, принятый в современных CASE-системах и ориентированный на методы управляемой моделими разработки, описанные выше. Поведение объектов характеризует действия, когда объект либо является инициатором переноса информации от него к другому объекту, либо подвергается аналогичному воздействию от других объектов. В процессе поведения объекта изменяется его состояние, что отображается в массивах контекстов состояний объектов. Иерархическая модель объектов позволяет применять псевдопараллельные методы

вычислений характеристик объекта, начиная с нижних уровней иерархии к верхним, это позволяет сократить время расчета и приблизить его к реальному времени. При сохранении проекта сохраняются все состояния отдельных объектов модели, и при последующей загрузке возможно продолжение работы или модификация составленной модели.

Таким образом, сформулируем постулаты методики моделирования в разработанной среде.

1. Существуют объекты, предназначенные для моделирования отдельных подсистем ИС.
2. Объект обладает совокупностью свойств, полей и методов, проявляющихся во взаимодействии объектов с объектами, пространством и временем, то есть выполняется принцип инкапсуляции.
3. Дерево классов построено в соответствии с иерархией: каждый класс-потомок наследует поля, свойства и методы класса-родителя и может иметь собственные поля, свойства и методы – принцип наследования.
4. Классы реализуют схожие по смыслу задачи разными способами посредством специальных функций (алгоритмов) – принцип полиморфизма.
5. Действия выражаются изменением свойств объекта, по сути являющихся переменными.
6. Взаимодействия – это совокупные изменения значений нескольких информационно связанных переменных. Так моделируется перенос информации.
5. Совокупность моментальных «снимков» состояний всех переменных, относящихся к модели, – это поведение модели или пространство ее состояний.
6. Свойства объекта формализуются математическими моделями и вычислимы.
7. Принимается, что конкретный экземпляр объекта находится в пространстве и во времени. Время абстрактно, но имеет начало и конец, его единицы могут быть любыми; время разделяется на такты – минимальные единицы времени существования и поведения объектов. Существует временной интервал моделирования.

Разработка программного комплекса моделирования ИС на основе изложенной методики позволила реализовать все основные принципы объектно-ориентированного программирования, что дало основание считать разработанную визуальную среду полностью объектно-ориентированной и отличающейся от известных.

Литература

1. Бутакова М.А. Модели информационных потоков в системах массового обслуживания на транспорте: монография. Рн/Д: Изд-во РГУ, 2006. 228 с.
2. Butakova M., Klimanskaya E., and Chernov A. Fast Algorithm for Simulation of Levy Stable Stochastic Self-Similar Processes. Proc. 5th Intern. Conf. Intelligent Systems, Modelling and Simulation (ISMS), pp. 225–229, 27–29 Jan. 2014. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7280860> (дата обращения: 12.06.2014).
3. Бутакова М.А. Исследование телекоммуникационных сетей в условиях автомобильных потоков с сильным последствием // Изв. вузов. Сев.-Кав. регион. Сер. Техн. науки. 2006. № 4. С. 18–23.
4. Gordon G. A general purpose systems simulation program. Proc. EJCC. Washington, NY: Macmillan Publ. Co., Inc. 1961, pp. 87–104.
5. Gordon G. A general purpose systems simulator. IBM Systems J. 1962, vol. 1, no. 1, pp. 18–32.
6. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004. 320 с.
7. Лазарев Ю. Моделирование процессов и систем в MATLAB: учеб. курс. СПб: Питер; Киев: BHV, 2005. 512 с.
8. OMG Unified modeling language specification. URL: <http://www.ibm.com/software/rational> (дата обращения: 12.06.2014).
9. ITU Recommendation Z.100: Specification and Description Language. 1993, 204 p.
10. Бардзинь Я.М., Калкинъш А.А., Стродс Ю.Ф., Сыцко В.А. Язык спецификаций SDL/PLUS и его применения. Рига, 1988. 313 с.
11. Карабегов А.В., Тер-Микаэлян Т.М. Введение в язык SDL. М.: Радио и связь, 1993. 184 с.
12. ITU Recommendation Z.200 – High Level Language (CHILL). 1993.
13. ITU-T MSC2000R3 Draft Z.120(11/99) Message Sequence Charts ITU-T Recommendation Z.120.
14. ITU Recommendation Z.100 – Appendices I and II: SDL Methodology Guidelines, SDL Bibliography. 1993, 129 p.