

МЕТОДОЛОГИЯ, ТЕХНОЛОГИЯ И ПРИНЦИПЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ ИМИТАЦИОННЫХ ПРИЛОЖЕНИЙ

С. А. Власов (Москва), В. В. Девятков (Казань), Н. Б. Кобелев (Москва)

Современный уровень развития информационных технологий, теоретический базис имитационных исследований и накопленный опыт разработчиков позволяют в настоящее время расширить и дополнить классический подход к процессам создания и внедрения имитационных моделей.

Еще в 60–70 годы XX века Бусленко [1–2], Шенноном [3], Нейлором [4] и другими классиками имитационного моделирования был заложен фундамент теории имитационных исследований, в это же время появились и основные языки имитационного моделирования. По существу, весь этот базис актуален и в настоящее время. Да, появляются новые алгоритмы, совершенствуются средства взаимодействия пользователя с моделью, автоматизируются те этапы исследований, которые ранее были ручными, но все же имитационное исследование больше остается наукой и искусством, нежели инженерным средством для огромной армии системных аналитиков и инженеров. Имитация до сих пор остается делом профессионалов «симуляционистов», и это сдерживает ее огромные потенциальные возможности использования в реальной экономике. Появление и внедрение коммерческих симуляторов – достаточно успешный шаг в расширении применимости метода имитационного моделирования, однако их распространение ограничено высокой стоимостью и отсутствием русскоязычного программного обеспечения.

Авторами работы (группа специалистов ИПУ РАН, ВЗФЭИ и ООО «Элина-Компьютер») предлагается несколько иная методология имитационных исследований, основанная на интегрированной технологии и едином программном базисе проведения исследований. Главная цель данного подхода – сделать имитационное моделирование интуитивно понятным и легким в освоении инженерным средством анализа сложных систем для широкого круга специалистов.

В основе предлагаемой методологии и инструментальных средств лежат два главных принципа:

доступность и возможность для пользователя проведения в рамках единой программной системы всех классических этапов имитационных исследований;

возможность работы с данными инструментальными средствами для максимально широкого круга специалистов посредством простого и удобного конструирования исследуемой системы из стандартных «кирпичиков».

Программным результатом применения данной методологии, в каждом конкретном случае, должен быть единый программный комплекс – имитационное приложение. Причем имитационное приложение должно быть «заточено» под данную проблему, «говорить» на языке пользователя, решать значимые и конкретные, а не гипотетически важные задачи.

С теоретической точки зрения реализация имитационного приложения состоит в использовании единой методики формализации сложных дискретных систем, разработке языка представления исследуемых систем пользователем и преобразовании данного представления в имитационную модель.

С практической точки зрения имитационное приложение – это программная реализация данных теоретических основ и технологии имитационных исследований – создание интегрированной программной среды с использованием уже хорошо отработанных и известных программных средств симуляторов, пакетов обработки статистики,

программ мониторинга данных, систем планирования экспериментов и даже офисных программ.

Структура имитационного комплекса моделирования

Рассмотрим вариант комплексного решения создания имитационного приложения в рамках интегрированной технологии и единого программного базиса в виде единого имитационного комплекса моделирования (ИКМ), состоящего из языка пользователя (ЯАП), типовой имитационной процедуры (УИМ-1) и программного базиса (ПБ).

Структура ИКМ (рис. 1) состоит из трех основных частей: блока пользователя, внутреннего блока ИКМ и управляющей программы. Каждая часть состоит из ряда подсистем, причем некоторые подсистемы используются совместно.

Процесс имитационного исследования при использовании ИКМ должен начинаться с деятельности блока пользователя. Под пользователем понимается специалист-аналитик или группа специалистов предприятия, к которому относится объект моделирования, которым поручено построение модели данного объекта. Будем далее данную группу пользователей называть аналитиками.

Аналитики должны обладать глубокими профессиональными знаниями по данному объекту, представлять задачи, которые должна решать модель объекта, владеть языком пользователя объекта ЯАП и в полной мере понимать суть и освоить практически все другие программные средства ИКМ. Владение языком пользователя ЯАП и технологией работы в ИКМ предполагает: общепринятое владение компьютером, прохождение краткосрочных курсов по изучению ИКМ и ЯАП, в частности. Данный курс не должен требовать изначально знания каких-то языков программирования или специальной математической подготовки.

Исходя из своего опыта в отношении деятельности объекта моделирования, аналитик задает структурную схему модели, воспроизводящую последовательность операций, которую нужно промоделировать. Имитационное моделирование предполагает воспроизведение объекта моделирования таким образом, что последовательность и особенности его функционирования фиксируются в модели без искажения или деформации материальных потоков или потоков информации и тех мест (блоков), где проходит их обработка. Имитация объекта должна быть максимально полной, а материальные потоки или потоки информации должны быть замещены их модельными представлениями с минимально допустимой погрешностью для конкретных типов производств [3, 5–9].

Построение структуры и логики функционирования модели для аналитика на ЯАП должно начинаться с применения «Подсистемы задания, структуры и логики модели», в рамках которой описываются все типовые блоки модели, логика взаимосвязи их параметров, состояний, входных и выходных сигналов, а также матрица сопряжения блоков модели между собой. В ЯАП разработчики ИКМ создают иерархию типовых блоков по каждой области исследования.

После задания структуры и логики, также на ЯАП, аналитик формулирует и вводит в компьютер в числовой, табличной или графической форме все входные данные, значения параметров и состояний.

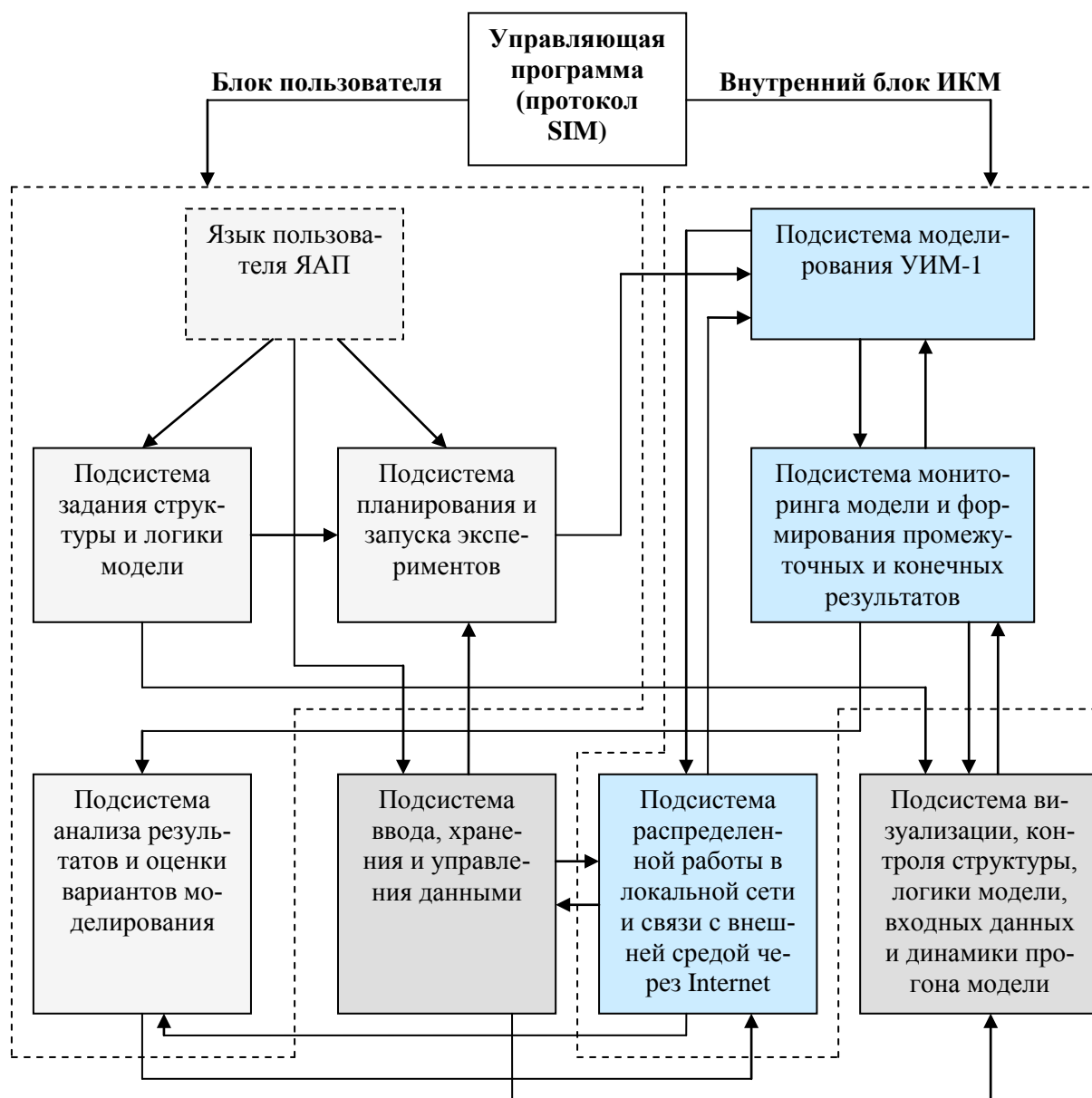


Рис. 1. Общая схема ИКМ

Входные потоки информации, как правило, должны представлять собой по характеристикам и моментам появления новых значений: «фотографии» аналогичных значений из прошлого, их предполагаемый вид из будущего, их желаемое значение или варианты, которые необходимо исследовать. Во многих случаях нет необходимости применения теоретических распределений при первоначальном формировании потоков данных. Эти функции выполняет «Подсистема ввода, хранения и управления данными». Кроме функций ввода данных она обеспечивает их хранение и управление, как для каждого конкретного эксперимента, так и для серии экспериментов. В рамках имитационного приложения данная подсистема представляет собой базу данных.

В рамках «Подсистемы визуализации, контроля структуры, логики модели и входных данных и динамики прогона модели» обеспечивается отображение структуры

и логики модели на экране компьютера, осуществляется контроль: правильности введения логики взаимосвязей, параметров блоков модели, их числовых значений, матрицы сопряжения блоков модели, входных потоков и т. п. Также данная подсистема обеспечивает визуализацию динамики хода прогона имитационной модели на основании данных мониторинга модели. Анализируя ход прогона, аналитик может досрочно остановить прогон и перейти к анализу текущих результатов.

«Подсистема планирования и запуска экспериментов» ЯАП обеспечивает ввод данных для окончательного формирования прогона одного или серии экспериментов. Причем этот ввод аналитик может осуществлять либо вручную, либо автоматизированно, посредством запуска программ планирования экспериментов и оптимизации. Завершающим действием аналитика при работе с данной подсистемой является выдача команды – начать имитационный прогон.

В блоке пользователя находится также «Подсистема анализа результатов и оценки вариантов моделирования», к работе с ней аналитик переходит после завершения (или прерывания) имитационного прогона. Эта подсистема получает информацию от подсистемы мониторинга и обработки промежуточных и конечных результатов моделирования. Полученная статистическая, логическая или иная информация поступает пользователю для рассмотрения анализа и принятия решений о ее дальнейшем использовании в процессе данного имитационного исследования. Полученная информация может далее проходить и другие этапы обработки – экспертные, аналитические, регламентационные и т.п.

Таким образом происходит работа аналитика с ИКМ в процессе проведения имитационного исследования.

Рассмотрим теперь основные функции внутреннего блока ИКМ – блока, подсистемы которого не доступны аналитику при управлении ходом имитационного исследования аналитику, а доступны лишь результаты работы.

«Подсистема моделирования УИМ-1» – одна из главных частей модели, которая и обеспечивает процедуру имитации модели. После завершения формирования модели на языке ЯАП осуществляется генерация кода (создание скрипта) на одном из известных языков имитационного моделирования. В качестве «атомарного» языка имитационного моделирования может быть использован любой достаточно мощный и проверенный практикой язык. Важнейшей частью УИМ-1 является генератор модели системы по введенным аналитиком данным. Теоретический базис, используемый при программной реализации транслятора с ЯАП, позволяет генерировать корректный и адекватный код модели одного или серии экспериментов. Далее идет процесс имитационного прогона модели средствами выбранного при разработке языка имитационного моделирования. В процессе прогона реализуются все функции, присущие при имитации дискретно-событийных моделей. А в подсистеме «Мониторинга модели и формирования промежуточных и конечных результатов» осуществляется динамический мониторинг состояния модели и накопление результатов в соответствии с введенными в подсистему планирования экспериментов указаниями аналитика. Получение промежуточной информации хода моделирования может быть необходимо, если эта информация используется для оперативного вывода результатов, для анализа аналитиком и принятия решения о продолжении или прерывании имитационного прогона. Эта же подсистема ведет предварительную статистическую обработку результатов моделирования. При завершении имитационного прогона она передает ее подсистеме визуализации для наглядного отображения.

В структуре внутреннего блока ИКМ имеется также «Подсистема распределенной работы в локальной сети и связи с внешней средой через Internet» для реализации имитационных моделей открытого типа.

«Подсистема ввода, хранения и управления данными» и «Подсистема визуализации, контроля структуры, логики модели и входных данных» используются совместно как в блоке пользователя, так и в работе внутренних процедур ИКМ.

Управляющая программа обеспечивает взаимодействия всех подсистем комплекса ИКМ на основе специального протокола SIM, построенного на базе XML стандарта. Протокол SIM должен унифицировать обмен всеми типами данных, возникающими в процессе имитационного исследования. При завершении разработки планируется сделать данный протокол открытым и общедоступным для использования.

Теоретические основы предлагаемой методологии

Классическая схема построения имитационной модели сложного объекта предполагает следующие основные этапы: 1) содержательное описание объекта; 2) постановка задачи моделирования; 3) выбор метода моделирования; 4) формализация задачи; 5) построение модели, ее апробация на конкретном примере и ее уточнение; 6) моделирование и получение различных статистических характеристик; 7) анализ результатов моделирования; 8) варианты использования модели в практической деятельности. Первые пять этапов предполагают их реализацию на языке пользователя.

Рассмотрим общие подходы к анализу и структуризации сложных объектов для представления их в виде, удобном для моделирования на языке пользователя ЯАП. Приведем последовательность определений, конкретизирующих главный элемент ЯАП – понятие системы.

Система – это совокупность множества элементов (частей целого), множества связей и отношений порядка между ними, объединенных единой целью функционирования. Всякий моделируемый объект представляется в виде совокупности каких-то частей-элементов, которые можно представить в виде систем. Взаимодействия между этими частями осуществляются при помощи связей. Отношения порядка между частями означают приоритетность, или первенство каких-то одних частей по отношению к другим. Наиболее понятно отношения порядка просматриваются в иерархических системах, где каждый элемент является либо управляемым, либо управляющим. Формализуем эти понятия.

Предположим, что имеются некоторые конечные множества: множество элементов $a = (a_1, a_2, \dots, a_u)$, $k = \overline{1, u}$; множество связей $F = (f_1, f_2, \dots, f_w)$, $v = \overline{1, w}$ и множество отношений порядка $P = (P_1, P_2, \dots, P_z)$, $z = \overline{1, z}$, заданные на некотором пространстве Γ , т. е. $a \subset \Gamma, F \subset \Gamma, P \subset \Gamma$. Элементы перечисленных множеств являются проекциями некоторой точки O из Γ на одноименные оси пространства Γ ,

$$O = (a_1, a_2, \dots, a_u; f_1, f_2, \dots, f_w; p_1, p_2, \dots, p_z).$$

Будем интерпретировать точку O как некоторую организацию, состоящую из множества элементов a , связей F , отношений порядка P . Зададим теперь некоторый функционал G , зависящий от a, F, P , т.е.

$$G = g(a, F, P),$$

который определяет некоторый порядок взаимодействия множеств a, F, P между собой и преобразует организацию O в некоторую систему A , т.е. $O \xrightarrow{G} A$. Таким образом, под системой A понимается организация O , преобразованная функционалом G . Функционал G может интерпретироваться содержательно, как целевая функция системы A .

Если множества a, F, P подлежат какой-либо модификации (дополняются новыми составляющими либо некоторые из них ликвидируются или преобразуются) и при этом функционал G остается неизменным, то говорят, что система A изменила свою структуру, или

$$G = g(a, F, P) = g'(a', F', P') = const,$$

где (\prime) обозначает модифицированное значение.

Теперь некоторую систему A можно записать в виде

$$A = \{a, F, P, G\},$$

а ее инвариант – структуру в виде

$$A' = \{a', F', P', G\}.$$

Количество структур A' системы A в общем случае неограниченно.

Если в системе A изменяется функционал G , это означает, что данная система преобразовалась в другую систему \bar{A} , не эквивалентную системе A .

Таким образом, мы провели структуризацию некоторой системы A , позволяющую осуществлять различные анализы этой системы при условии, что ее атрибуты a, F, P, G принимают вполне определенный вид для конкретного реального объекта.

Рассмотрим теперь способы формализации A -систем. Представим элемент a_k системы A в виде схемы, представляющей агрегат Н.П. Бусленко, где $x = (x_1, x_2, \dots, x_n)$, $y = (y_1, y_2, \dots, y_m)$, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_p)$, $s = (s_1, s_2, \dots, s_p)$ – соответственно входные, выходные клеммы, параметры и состояния агрегата a_k . Агрегат a_k является элементарным блоком системы A . Объединение нескольких агрегатов через входы и выходы образует систему A .

A -система, состоящая из агрегатов a_k – имеет множество связей F , которые будем обозначать через множество пар $F = \{y_j^k x_i^{k+l}\}$, где каждая пара $(y_j^k x_i^{k+l})$ обозначает связь между выходной клеммой y_j^k и входной клеммой x_i^{k+l} . Множество отношений порядка P может быть задано различными способами (все элементы равноправные, и сигналы поступают в соответствии с блок-схемой A -системы и номером агрегата a_k , приоритет определяется специальным агрегатом, приоритет задается иерархически и т. д.). Целевая функция G_k для каждого агрегата a_k определяется в виде определенного функционала $G_k = (\alpha^k, s^k)$. Для A -системы целевая функция может быть записана как $G = g\{(\alpha_k), (F_k), (G_k)\}$, хотя получить конкретную зависимость ее для какой-либо модели вряд ли удастся. Вообще для сложных объектов или их имитационных моделей целевая функция может быть определена по характеристикам агрегатов, описывающих главные (генетические) функции моделируемого объекта. Целевые функции всех остальных агрегатов данной A -системы должны поддерживать генетические функции и тем более не противодействовать им.

Таким образом, A -система полностью отвечает введенным определениям системы, организации, структуры.

Дальнейшая формализация A -системы связана с уточнением способов описания x, y, α, s . Будем теперь называть агрегат a_k элементарным блоком.

Зададим структуру входного сигнала $x_i^k(t)$ – в виде некоторого набора $(k, i, \pi_1, \pi_2, \dots, \pi_{i_k}, t)$, а выходного сигнала $y_j^k(t)$ – в виде аналогичного набора $(k, j, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{j_k}, t)$, где k – номер типового блока; i – номер входной клеммы; j – номер выходной клеммы; $(\pi_1, \pi_2, \dots, \pi_{i_k})$ – характеристики входного сигнала по i -му входу; $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{j_k})$ – характеристики выходного сигнала по j -му выходу; t – момент поступления сигнала.

При описании входных или выходных сигналов и состояний могут быть использованы любые, ничем не ограниченные математические конструкции. Это могут быть числа, множества, функционалы, логические или иные фразы, графические изображения, правила и т.п. Состояния и параметры элементарных блоков также имеют определенную структуру. Условимся задавать состояние типового блока в виде совокупности $(k; S_1, S_2, \dots, S_{p_k}; \alpha_1, \alpha_2, \dots, \alpha_{p_k}; t)$, где k – номер типового блока; S_1, S_2, \dots, S_{p_k} – характеристики состояния блока k ; $\alpha_1, \alpha_2, \dots, \alpha_{p_k}$ – параметры блока k ; t – текущее время.

Семантическая схема ЯАП

Сфера применения имитационного моделирования в настоящее время гораздо шире обычных научных исследований. Потенциально оно может быть использовано в таких областях практической экономики, как системные исследования бизнеса, технологий и новых решений в любой средней и крупной компании, и должно стать важным инструментом при принятии крупных и ответственных системных решений на предприятии. Поэтому существенно возрастают требования к построению интерфейса взаимодействия пользователя с моделью (в нашем случае это ЯАП). Необходимо выбрать семантическую схему построения языка, интуитивно понятную практически любому грамотному человеку из любой сферы деятельности с минимальными требованиями к дополнительному обучению. Вместе с тем эта схема должна быть удобной для формализации и дальнейшей программной реализации.

Итак, мы сформировали язык А-систем, который назовем языком А-систем пользователя (ЯАП). Элементарным блоком «кирпичиком» этого языка является А-система или типовой элементарный блок (ТЭБ). Количество ТЭБ в языке не ограничено. Для каждой конкретной области должны иметься наработанные стандартные ТЭБ. Т. е. должна быть некая иерархия ТЭБ. Кроме этого аналитик из имеющихся элементарных блоков может «конструировать» свои ТЭБ.

Для облегчения освоения ЯАП аналитиком следует типизировать функции ТЭБ, чтобы пользователь не настраивал каждый раз элементарные блоки для своей модели, а брал готовые с уже реализованными функциями. Поэтому для ЯАП должен быть разработан своеобразный «алфавит» типовых ТЭБ. ТЭБ может быть заранее настроен на выполнение определенных функций от элементарных (сложение, функциональное преобразование, логические действия, система массового обслуживания, запаздывание, переключения связей и т. п.) до сложных. Сложные функции представляют собой, например, систему автоматического управления; функциональный накопитель (склад любого типа); типовой технологический процесс (непрерывный, дискретный, сборочный, комплектующий, обрабатывающий и т. п.); типовой производственный процесс (цеха, участка, стройки и т. п.); типовое предприятие со всеми функциями и службами и т. п.

Учитывая различную степень сложности и специфичности моделируемых объектов, разобьем их на категории и распределим ТЭБ различных типов в общей библиотеке ЯАП, например, по следующим категориям:

первая категория (К1) соответствует ТЭБ, не ограниченному какими-либо зависимостями или условиями, т. е. представляет собой ТЭБ самого общего вида, кото-

рый может быть настроен пользователем на выполнение любого алгоритма, не содержащегося в библиотеке ЯАП;

вторая категория (К2) содержит ТЭБ, выполняющие самые элементарные функции, т. е. простые арифметические и логические действия, вычисления элементарных функций, функциональные преобразования и т. п.;

третья категория (К3) обеспечивает моделирования входных потоков заданного вида (теоретических и эмпирических), систем массового обслуживания, сложных преобразователей и т. п.;

четвертая категория (К4) моделирует иерархические структуры, списки, файлы и т. п.;

пятая категория (К5) воспроизводит простые объектно-ориентированные структуры, например, моделирование отдельных элементов технологического процесса, предприятия (склад, цех, участок, конвейер, информационная система и т. п.) и т. д. и т. п.

Всего на сегодняшний день принято деление ТЭБ на десять категорий.

Графические средства ЯАП позволяют быстро и наглядно конструировать логическую структуру модели из ТЭБ.

Также в языке предусмотрены диалоги по вводу числовых данных о ТЭБ, логике их соединений, планированию экспериментов и т. д. ЯАП обеспечивает определенную типизацию и формализацию таких процедур, что позволяет существенно упростить проведение имитационных исследований аналитиком. Формализация имитационных процедур должна предусматривать все основные способы их организации, а при появлении новых идей и способов адаптировать их.

Практическая апробация методологии

В настоящее время проведен лишь первый этап создания методологии, определены границы, выбран теоретический базис, сформулированы принципы технологии имитационных исследований. Предстоит провести кропотливую детальную работу по многим направлениям. Однако уже сейчас можно привести ряд практических примеров, подтверждающих правильность выбранного подхода [7–9]. Это ряд имитационных приложений, разработанных в процессе работы над методологией:

- модель центра коллективной обработки Банка России;
- модель системы сервисного обслуживания вычислительной техники в ОАО «Татнефть»;
- укрупненная модель железнодорожного направления;
- упрощенная модель торгового центра;
- модель технического обслуживания самолетов в авиакомпании;
- модель распределения ресурсов в агрохолдинге во время проведения сельхозкампаний.

Для каждого из этих приложений разработан ряд своих специализированных ТЭБ. Например, для модели железной дороги это – поезд, станция, перегон. Любая из этих моделей представляет собой сборку таких ТЭБ. Причем количество используемых ТЭБ зависит от того, какая задана структура модели и какие введены числовые параметры. Каждому ТЭБ в приложении соответствует настраиваемый шаблон на языке имитации. И после выдачи аналитиком команды начать имитационный прогон генератор на основе введенных данных, шаблонов и правил их объединения создает исполняемый код модели. Примеры конкретной визуализации при реализации ЯАП в данных приложениях показаны на рис. 2. Таких диалоговых форм в каждом приложении очень много. Их количество определяется проблемной областью, степенью детализации, возможностями изобразительных средств и т. д.

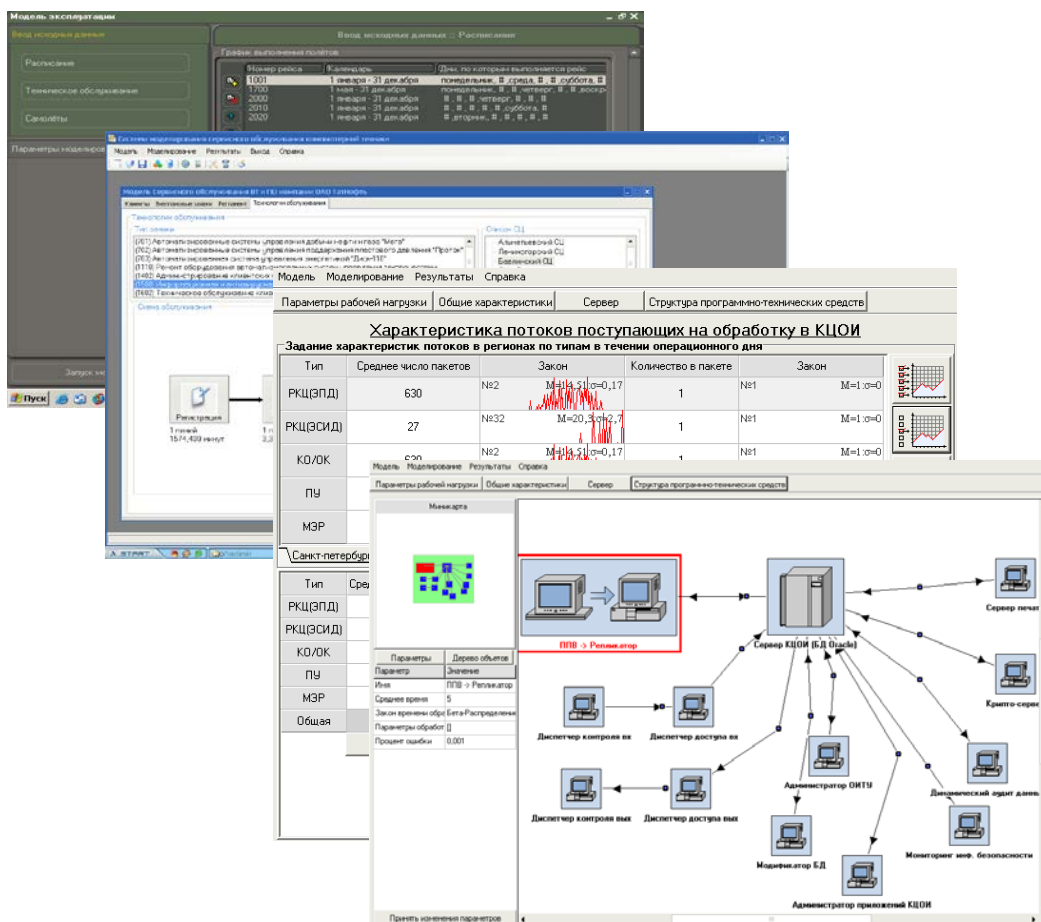


Рис. 2. Визуальные примеры реализации ЯАП в имитационных приложениях

Одним из основных общих признаков данных приложений является охват большинства этапов имитационного исследования в рамках единого программного продукта. Конечно, это лишь частные случаи применения данного подхода, и конструирование и программную реализацию данных приложений осуществляли специалисты по ИМ и программированию. Но в результате их реализации уже создан ряд общих для большинства имитационных исследований компонентов: графическая визуализация структуры, генератор моделей, мониторинг модели, взаимодействие с офисными компонентами и т. д.

В качестве подсистемы имитации, при разработке имитационных приложений был использован язык GPSS World [5]. Он в полной мере отвечает всем требованиям, как по функциональному наполнению и гибкости сборки кода модели из составных частей, так и по возможностям управления его вводом и выводом.

Выводы

1. Существенное повышение уровня компьютеризации, рост производительности персональных компьютеров и появление новых программных технологий открывают возможности для перевода имитационных исследований из категории «науки и искусства» в категорию инженерных средств.
2. Создан теоретический базис и сформулированы принципы методологии имитационных исследований в рамках единого программного базиса.
3. Осуществлена программная апробация методологии.

Литература

1. **Бусленко Н. П., Шарагина З. И.** Математическое моделирование производственных процессов на цифровой вычислительной машине. М.: Физматгиз, 1964.
2. **Бусленко Н. П.** Моделирование сложных систем. М.: Наука, 1968.
3. **Шеннон Р.** Имитационное моделирование систем – искусство и наука. М.: Мир, 1978. 418 с.
4. **Нейлор Т.** Машинные имитационные эксперименты с моделями экономических систем. М.: Мир, 1975. 500 с.
5. **Кобелев Н. Б.** Основы имитационного моделирования сложных экономических систем. М.: Дело, 2003, 335 с.
6. Руководство пользователя по GPSS World/Перевод с английского. Казань: Мастер Лайн, 2002. 384 с.
7. **Смирнов В. С., Власов С. А., Ваулинский Е. С., Лебедев Б. И.** Методы и модели управления проектами в металлургии. М.: СИНТЕГ, 2001.
8. **Самойлов В. В., Власов С. А., Девятков В. В.** Имитационное исследование системы сервисного обслуживания программно-технических средств ОАО «Татнефть»//Автоматизация в промышленности. 2007. № 4. С. 11–14.
9. **Девятков В. В.** Разработка приложений в среде GPSS WORD//Вторая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности. СПб.: ЦНИИТС, 2005.