

На правах рукописи



Карпухин Илья Николаевич

**Исследование и реализация  
систем дискретно-событийного имитационного  
моделирования на основе графовых моделей**

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Москва, 2013

Работа выполнена на кафедре моделирования систем и сетей федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Российский государственный социальный университет».

**Научный руководитель:**

**Кораблин Юрий Прокофьевич**

доктор технических наук, профессор, профессор кафедры моделирования систем и сетей федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Российский государственный социальный университет»

**Официальные оппоненты:**

**Фальк Вадим Николаевич**

доктор технических наук, доцент, профессор кафедры прикладной математики федерального государственного бюджетного образовательного учреждения высшего профессионального образования Национального исследовательского университета «Московский энергетический институт»

**Аникеев Александр Викторович**

кандидат технических наук, доцент, главный специалист ООО «КВАРЦ – Новые Технологии»

**Ведущая организация:**

**Федеральное государственное бюджетное учреждение науки Институт системного анализа Российской академии наук (ИСА РАН)**

Защита состоится «17» апреля 2013 г. в 15-30 на заседании диссертационного совета Д 212.131.05 при МГТУ МИРЭА по адресу:  
Москва, 119454, пр-т Вернадского, д. 78, Д412

С диссертацией можно ознакомиться в библиотеке МГТУ МИРЭА.

Автореферат разослан «16» марта 2013 г.

Отзывы на автореферат в двух экземплярах, заверенные печатью, просим направлять по адресу 119454, г. Москва, пр-т Вернадского, 78, диссертационный совет Д 212.131.05

Ученый секретарь  
диссертационного совета  
к.т.н, доцент



Е.Г. Андрианова

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы исследования.** Имитационное моделирование является мощным инструментом исследования поведения реальных систем. Методы имитационного моделирования позволяют собрать необходимую информацию о поведении системы путем имитационного эксперимента над её математической моделью. Полученная информация используется затем для изменения или проектирования новой системы.

Современное имитационное моделирование применяется практически для всех классов математических моделей систем. В практике исследования операций и принятия решений имитационные эксперименты находят широкое применение в задачах создания систем связи; экономических задачах, включая оценку поведения потребителя, согласование цен, прогнозирование результатов деятельности фирм; задачах социологии и психометрии; задачах анализа военных стратегий и тактик, управления дорожным движением, логистики, динамики экосистем, социальных сетей и др.

В перечисленных выше областях одна из главных проблем создания имитационных моделей – выбор способа представления знаний, описывающих правила поведения системы, если модель не сводится к некоторой системе уравнений, допускающей аналитическое или численное решение. Отметим также необходимость учёта модели времени.

Адекватный выбор способа представления знаний минимизирует трудозатраты на создание модели и эффективность её дальнейшего использования. Например, широкий класс явлений и ситуаций могут быть описаны как системы массового обслуживания в дискретном времени, другими примерами моделей могут служить производственные модели принятия решений, сети Петри с их расширениями, байесовские сети и т.п.

В данной работе проводится анализ различных способов представления знаний, их преимуществ и недостатков с точки зрения имитационного моделирования, предлагается авторский вариант представления знаний на основе графовых моделей и реализуются инструментальные программные средства дискретно-событийного моделирования. Акцент делается на поиске компромисса между простотой создания и понимания пользователем представления знаний с одной стороны и мощностью этой модели с другой стороны.

**Объект работы** – модели дискретно-событийного имитационного моделирования, базы знаний на их основе.

**Предмет работы** – создание и интерпретация моделей поведения дискретных систем, эффективное манипулирование моделями.

**Цель работы** – повышение эффективности процессов создания моделей поведения систем и программ, разработка программного комплекса, используемого для повышения эффективности компьютерного

моделирования на основе баз знаний, хранящих информацию о поведении систем.

**Решаемые задачи.** Для достижения поставленной цели в работе решаются следующие задачи.

1. Исследование подходов к имитационному моделированию, моделей представления знаний и современных программных средств имитационного моделирования, выявление их достоинств и недостатков.

2. Разработка авторской модели представления знаний для имитационного моделирования систем в дискретном времени, позволяющей охватить широкий круг как прикладных, так и учебных задач. Разработка алгоритмов интерпретации, верификации и эквивалентного преобразования модели.

3. Реализация модели и алгоритмов в виде программных библиотек общего назначения.

4. Исследование разработанных алгоритмов и их реализаций, заключающееся в доказательстве корректности, установлении границ применимости.

5. Создание на основе библиотек прикладного программного комплекса *G-IPS Ultimate* для интерактивного манипулирования моделями представления знаний и проведения имитационных экспериментов.

6. Применение программного комплекса *G-IPS Ultimate* для получения теоретических и прикладных результатов, использование в учебном процессе.

**Методы исследований и достоверность результатов.** Задачи, поставленные в работе, решаются с помощью методов теории множеств, математической логики, теории графов, теории вероятностей и математической статистики, анализа и построения эффективных алгоритмов и др.

Достоверность результатов обосновывается доказательством свойств предложенных моделей и подтверждается объёмными вычислительными экспериментами.

**Теоретическая значимость.** В работе представлены следующие новые научные результаты.

1. На основе проведённого анализа существующих способов представления экспертных знаний, их достоинств, недостатков и области применения, а так же исследования подходов к имитационному моделированию (дискретно-событийное, системно-динамическое, многоагентное) предложена и формализована оригинальная математическая модель, формализующая представление экспертных знаний для имитационного моделирования систем в дискретном времени. Она представляет собой набор графовых моделей (наглядных *решающих графов*), обобщает подход на основе продукционных правил и позволяет охватить

широкий спектр прикладных и учебных задач при сохранении простоты интерпретации, присущей продукционным моделям.

2. Предложен и реализован алгоритм, позволяющий выявлять такие свойства предложенной модели как полнота и неоднозначность, что позволяет оптимально обрабатывать модель.

3. Доказана сводимость многодольных решающих графов к функционально-эквивалентным трёхдольным, что позволяет обеспечить минимальное время цикла интерпретации модели, и предложен алгоритм такого сведения.

4. Проведены объёмные вычислительные эксперименты для получения оценки эффективности использования свойства сводимости при интерпретации модели, подтверждена корректность и получены оценки вычислительной сложности процесса интерпретации модели.

**Практическая полезность.** Разработанные модели и программные средства могут быть использованы для моделирования алгоритмов с применением методов параллельных вычислений, а так же для моделирования систем массового обслуживания. Помимо вышеназванных, важным направлением использования разработанных программных средств является решение задач экономики, логистики и задач учебного назначения.

Оригинальный способ представления знаний позволяет строить и обрабатывать базы знаний, содержащие модели поведения сложных систем из различных областей науки и техники. Такие модели могут использоваться для построения тренажеров (симуляторов), с помощью которых можно обучать оперативно-диспетчерский персонал.

Предложенная модель реализована в виде алгоритмов и программных библиотек, позволяющих создавать, верифицировать и тестировать имитационные модели, проводить имитационные эксперименты с целью определения свойств моделируемой системы, интерпретировать и анализировать результаты прогона моделей, визуализировать процесс имитации. Подтверждена корректность и функциональная полнота программных библиотек.

Разработан и апробирован программный комплекс *G-IPS Ultimate*. Данный комплекс можно использовать как при решении прикладных задач, так и в учебном процессе: при проведении лабораторного практикума по таким дисциплинам как исследование операций и принятие решений, теория игр, экспертные системы, имитационное моделирование, интеллектуальные системы, при выполнении курсовых и дипломных проектов.

**Внедрение.** К настоящему моменту разработанный программный комплекс внедрён ФГБУ «Федеральный Научно-Клинический Центр Детской Гематологии Онкологии и Иммунологии имени Дмитрия Рогачева» министерства здравоохранения РФ для моделирования бизнес-процессов, а

также в учебный процесс Российского государственного социального университета.

**Личный вклад диссертанта** состоит в разработке оригинального метода представления моделей представления знаний на базе продукционных правил, программной реализации алгоритмов трансформации и интерпретации этих моделей, а также в создании универсальных программных средств, позволяющих выполнять весь спектр необходимых задач, связанных с имитационным моделированием систем.

**Апробация работы.** Основные положения и результаты диссертации докладывались и обсуждались на 13 международной научно-технической конференции студентов и аспирантов «РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА» (г. Москва, 2007 г.), научно-практической конференции студентов, аспирантов, молодых ученых и специалистов: «Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте» (г. Коломна, 2009 г.), на научных семинарах РГСУ и НИУ ВШЭ.

**Структура и объём работы.** Диссертация состоит из введения, четырёх глав, заключения, списка литературы (117 наименований) и 3 приложений. Общий объём диссертационной работы составляет 142 страницы в основной части и 179 страниц приложений.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

**Во введении** обосновывается актуальность темы диссертации, формулируются цели и задачи исследования, научная новизна, теоретическая и практическая значимость работы.

**В первой главе** проведён обзор источников в области имитационного моделирования (ИМ). Систематизированы задачи и этапы ИМ: создание и верификация имитационной модели, проведение и визуализация имитационного эксперимента, анализ результатов эксперимента.

При изучении задач ИМ очень важно понимать то, как учитывается время в конкретной задаче. Оно может быть дискретным (например, при моделировании систем, основанных на системах массового обслуживания (СМО)), непрерывным (например, для задач физического моделирования) или не учитываться (например, для задач статистического моделирования). Чтобы определить круг наиболее актуальных задач, были исследованы модели времени в ИМ: дискретно-событийное моделирование, непрерывное моделирование, моделирование Монте-Карло. Обоснован выбор дискретного представления времени, так как данная модель позволяет охватить наиболее широкий (по сравнению с непрерывным моделированием) круг актуальных задач ИМ (моделирование систем, основанных на СМО, моделирование

бизнес-процессов, решение экономических и коммерческих задач, задач планирования, логистики и др.), а модели Монте-Карло могут быть представлены как частный случай дискретных моделей.

Помимо определения учёта времени при моделировании, важно то, какие цели преследуются разработчиком. Это может быть необходимость прояснения внутренней сути системы (выявление математических закономерностей), необходимость спрогнозировать поведение системы во времени или что-то другое. В зависимости от целей, различают и подходы к ИМ: многоагентный, дискретно-событийный подход и системная динамика. Для описания децентрализованных систем удобнее использовать методы многоагентного подхода, в то время как методы дискретно-событийного подхода позволяют описывать как централизованные системы, так и децентрализованные.

На основе проведённого анализа наиболее известных моделей представления знаний (формальных грамматик, логических, комбинаторных, алгебраических и продукционных моделей, семантических сетей, байесовских сетей, фреймов, ленемов, нечётких логик, нейронных сетей и генетических алгоритмов) выявлено, что для создания достаточно универсальной системы, позволяющей строить модели для широкого круга прикладных и учебных задач (например, задач планирования, моделирования систем основанных или представимых в виде СМО, задач диагностики и др.) наиболее подходящей является модель на основе продукционных правил (которая, в отличие от фреймов, выгодно отличается простотой построения). Несмотря на это, было установлено, что вне зависимости от преимуществ и недостатков той или иной модели, многое зависит от конкретной программной реализации этой модели. Зачастую в этих реализациях недостаточно средств отладки и верификации построенных моделей, таких как специальные режимы тестирования.

**Во второй главе** проводится анализ наиболее известных и широко используемых программных средств ИМ с целью изучения возможностей, предоставляемых конкретными продуктами, их доступности, преимуществ и недостатков.

Было рассмотрено и систематизировано более 20 наиболее известных и широко используемых программных средств имитационного моделирования, таких как:

- средства визуального моделирования широкого назначения (например, *AnyLogic by XJ Technologies, SIMUL8*);
- системы визуального моделирования узкого назначения (*COMNET III by CACI Products* – моделирование информационных сетей, *MedModel by PROMODEL Corporation* – моделирование больничных комплексов);
- специализированные языки программирования (*GPSS by Minuteman Software, SIMNET II by Hamdy A. Taha*);

- библиотеки, используемые для ИМ на основе языков программирования общего назначения (*Ascape by The Brookings Institution (Java)*, *SimPy by SimPy Developer Team (Python)*).

Особый акцент сделан на рассмотрение программных средств ИМ, используемых в учебном процессе в российских ВУЗах: РДО (разработанное в МГТУ им. Баумана) и *SIMPR-Windows* (разработанное в НИУ МЭИ).

Проведённое исследование показало, что наиболее ранние средства преимущественно представляют собой низкоуровневые средства моделирования и особые языки программирования, которые хоть и отличаются полезными инструментами для моделирования, малопригодны для решения современных задач в виду сложности синтаксиса. Большинство современных средств являются высокоуровневыми средствами визуального проектирования или интегрированными средами разработки с поддержкой современных языков программирования, как универсальных (например, *Java*, *C++*, *Python*) так и предметно-ориентированных (*GPSS*, *Omola*). В российской высшей школе при изучении дисциплин связанных с интеллектуальными системами в большинстве случаев используются малоизвестные программные средства, разработанные на кафедрах, и ориентированные на узкий круг задач.

На основе сравнительного анализа программных средств ИМ выявлено, что на сегодняшний день недостаточно средств визуального низкоуровневого проектирования имитационных моделей, содержащих инструменты, которые позволили бы разработчику детально исследовать не только результаты проведения имитационного эксперимента, но и сам процесс имитации в каждый момент времени, что позволило бы скорректировать модель в процессе её отладки.

**В третьей главе** предлагается авторская модель представления экспертных знаний на основе продукционных правил, задаваемых в виде графовых моделей особой топологии (*решающих графов*).

Предлагаемая модель представления знаний расширяет область определения продукционных правил, с помощью алгебры  $L = \langle \Omega, +, \sim \rangle$ , где  $\Omega = \{T, F, E\}$  – множество из трёх элементов ( $T$  – «истина»,  $F$  – «ложь» и  $E$  – «ошибка»), «+» – коммутативная операция суммирования ( $+: \Omega^2 \rightarrow \Omega$ ) и « $\sim$ » – некоммутативная операция квазисравнения ( $\sim: \Omega^2 \rightarrow \Omega$ ). Эти операции не полны, но удобны для реализации процесса интерпретации модели, описываемой далее. Определение этих операций задаётся аналитически.

Операция суммирования:

- $x_1 + x_2 = T \Leftrightarrow x_1 = x_2 = T$ ,
- $x_1 + x_2 = F \Leftrightarrow (x_1 \in \{T, F\} \text{ и } x_2 \in \{T, F\} \text{ и } x_1 \neq x_2) \text{ или } x_1 = x_2 = F$ ,
- $x_1 + x_2 = E \Leftrightarrow x_1 = E \text{ или } x_2 = E$ .

Операция квазисравнения:



- $x_1 \sim x_2 = T \Leftrightarrow x_1 = x_2$ ,
- $x_1 \sim x_2 = F \Leftrightarrow x_1 \neq x_2$  и  $x_1 \neq E$ ,
- $x_1 \sim x_2 = E \Leftrightarrow x_1 \neq x_2$  и  $x_1 = E$ .

Таблицы истинности для данных операций приведены в таблицах 1 и 2.

Таблица 1. Таблица истинности операции суммирования

$x_1 \backslash x_2$	<b>T</b>	<b>F</b>	<b>E</b>
<b>T</b>	T	F	E
<b>F</b>	F	F	E
<b>E</b>	E	E	E

Таблица 2. Таблица истинности операции квазисравнения

$x_1 \backslash x_2$	<b>T</b>	<b>F</b>	<b>E</b>
<b>T</b>	T	F	F
<b>F</b>	F	T	F
<b>E</b>	E	E	T

Расширение области определения правил и переход к трёхзначной логике является актуальным потому, что в реальной системе датчик (определяющий состояние элемента системы) может отказаться, не ответить, показать неправдоподобное значение. Введение дополнительных датчиков, сигнализирующих об ошибках неинтуитивно и ненаглядно для разработчика. Введение «ошибочного» значения позволяет не только наглядно описывать поведения системы в таких ситуациях, но и корректно обрабатывать такие ситуации.

Предлагается описывать систему правил в виде набора решающих графов  $RGS = \{RG^t\}$ .

$RG^t = \langle D, M, A, S \rangle$  –  $t$ -дольный конечный решающий граф ( $t \geq 3$ ), где:

- $D = (X_1, \dots, X_k)$  – вектор ( $k = t - 2$ ), в котором  $X_i = (q_{i,1}, q_{i,2}, \dots, q_{i,n})$  – непустой вектор датчиков  $i$ -го порядка,  $i \geq 1, n \geq 1$  – конечные,  $X_1$  – вектор простых датчиков (или датчиков первого порядка), значение которых задаёт функция интерпретации, присваивая вектор исходных значений,  $(X_2, X_3, \dots, X_k)$  – векторы (слои) датчиков более высокого порядка (второго, третьего и т.д.), их значения определяет функция интерпретации по значениям датчиков предыдущего порядка. Датчик  $q_{i,j}$  задается парой  $\langle val, Con \rangle$ ; где  $val \in \Omega$  – значение датчика;  $Con = \{\langle id, w \rangle\}$  – непустое множество пар  $\langle id, w \rangle$ , где  $id \in [1..|X_{i+1}|]$  для слоёв от 1 до  $(t-2)$  и  $id \in [1..|M|]$  для датчиков  $(t-1)$ -го слоя – индекс датчика более высокого порядка, т.е. дуга, связывающая датчик  $q_{i,j}$  с датчиком более высокого порядка,  $w \in \Omega$  – вес дуги.

Таким образом, датчик определяется своим значением  $val$ , и взвешенными связями с датчиками более высокого порядка  $Con$  – множество пар «связь-вес», где вес ребра означает ожидаемое значение; на рис. 1 эти веса обозначены цветом дуг;

- $M = (q_{t-1,1}, q_{t-1,2}, \dots, q_{t-1,z})$  – непустой вектор датчиков высшего порядка,  $z \geq 1$ . Датчик  $q_{t-1,j}$  отличается от датчиков более низкого порядка отсутствием логических связей, он определяется лишь

значением  $q_{t-1,j} = val \in \Omega$ . Так как первый индекс датчиков доли  $M$  всегда равен  $(t-1)$ , в дальнейшем будем его опускать;

- $A = (a_1, a_2, \dots, a_p)$  – непустой вектор действий,  $p \geq 1$ ;  $a_i$  – идентификатор действия (реально выполняемое действие определяется интерпретацией);
- $S$  – матрица размером  $|M| \times |A|$ , устанавливающая связь между датчиками высшего порядка и действиями. Элементом данной матрицы является либо пустой элемент (отсутствие связи) либо тройка  $s_{ij} = \langle ord, lv, rv \rangle$ , где
  - $ord$  – целое число, такое, что нуль означает отсутствие действия для данного правила, а любое другое число означает приоритет действия в наборе действий подлежащих выполнению.
  - $lv$  – левая допустимая граница вероятности (из отрезка  $[1, 100]$ );
  - $rv$  – правая допустимая граница вероятности (из отрезка  $[1, 100]$ ).

Рассмотрим обозначения весов дуг на решающем графе (рис. 1).

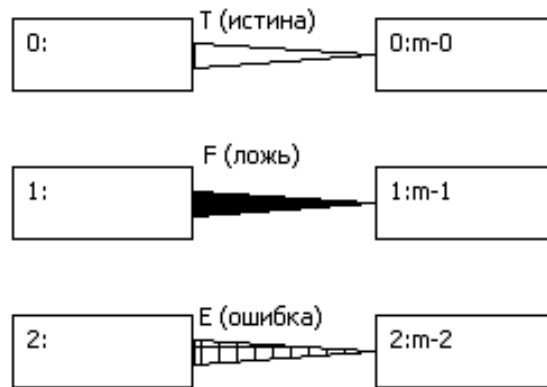


Рис. 1. Обозначения весов дуг в решающем графе

Вес дуги (ожидаемое значение датчика) обозначается графически так:  $T$  (истина) – зелёным или белым цветом в зависимости от режима отображения (цветной/монохромный),  $F$  (ложь) – красным или чёрным,  $E$  (ошибка) – серым или штрихованным.

На рис. 2 представлен пример решающего графа.

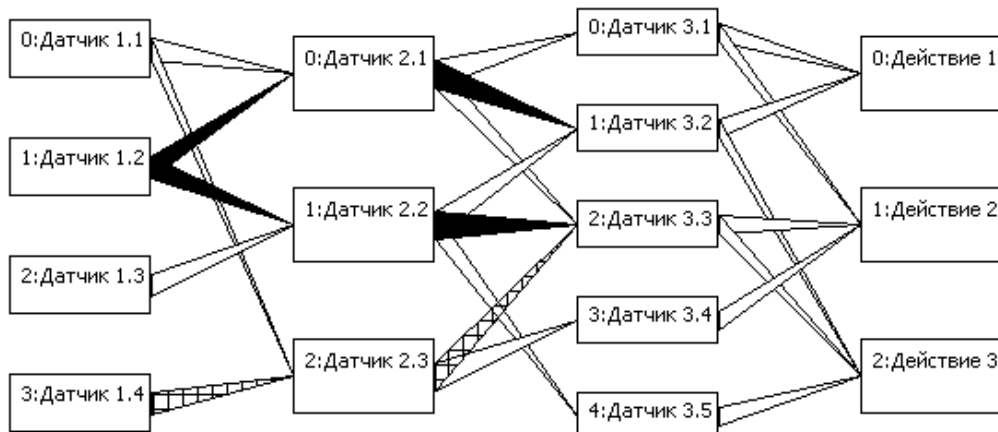


Рис. 2. Пример решающего графа

Множество  $D$  (рис. 2) содержит 2 множества  $X_1$  и  $X_2$ , где вектор  $X_1$  содержит датчики (1.1, 1.2, 1.3, 1.4), а  $X_2$  содержат датчики 2.1-2.3, множество  $M$  содержит датчики 3.1-3.5, и вектор действий  $A = (1, 2, 3)$ , а матрица  $S =$

	1	2	3
3.1	1 / 1 / 100	2 / 1 / 100	0
3.2	1 / 1 / 100	0	2 / 1 / 100
3.3	0	2 / 1 / 100	1 / 1 / 100
3.4	0	1 / 1 / 100	0
3.5	0	0	1 / 1 / 100

Рассмотрим процесс интерпретации решающего графа. Неформально, интерпретация – это некоторая функция, которая ставит в соответствие вектору *входных значений* упорядоченное мультимножество действий. Формально определим функцию интерпретации  $FI_{FP(RG^t, \xi)}: \Omega^{|RG^t.X_1|} \rightarrow B$ , где

- $RG^t$  – решающий граф;
- $\xi \in [1, 100]$  – случайная величина (для каждой итерации генерируется заново) – зерно (*random seed*);
- $B = \{B_i \in RG^t.A\}$ , – частично упорядоченное по приоритетам мультимножество элементов из  $A$ , где  $i \geq 0$ . То есть, на мультимножестве  $B$  определено отношение частичного порядка  $\rho$  такое, что:

$\forall a_j, a_{j+1} \in B \ a_j \rho a_{j+1}$ , где  $a_j, a_{j+1}$  получены из датчика  $q_k \Leftrightarrow S(M.q_k, a_j).ord \leq S(M.q_k, a_{j+1}).ord$ , где  $j \in [1..|B|]$ ;

- $FP(RG^t, \xi)$  есть тройка  $\langle FS, FQ, FA \rangle$ , где
  - $FS((v_h)): \{X_1.q_{1,i}.val\} \rightarrow \Omega$ ,  $i \in [1..|RG^t.X_1|]$ ,  $h = |RG^t.X_1|$  – функция, задающая значения датчиков первой доли; будем говорить, что функция  $FS$  применяет вектор  $(v_h)$  к датчикам первой доли, присваивая соответствующие значения;
  - $FQ = (f_{2,1}, \dots, f_{t-1,n})$  – система функций, где  $f_{(i+1),j}: \{q_{i+1,k}\} \rightarrow \Omega$ ,  $i \in [1..(t-2)]$ ,  $j \in [1..|X_{i+1}|]$ ,  $k \in [1..|X_i|]$ ,  $n = |M|$ ;
  - $FA = (r_1, \dots, r_n)$  – вектор процедур, заполняющих мультимножество  $B$  элементами из  $A$ , согласно матрице  $S$ ; где  $n = |M|$ .

$(v_h)$  – вектор *входных значений*,  $h \in [1..|RG^t.X_1|]$ , переменные со значениями из  $\Omega$ , определяемые вне интерпретации;

Функция  $FI$  ставит в соответствие любому вектору  $(v_h)$  мультимножество  $B$ , содержащее элементы из  $A$ .

Значение *val* датчиков, принадлежащих долям от 2 до  $(t-1)$ , вычисляется следующим образом:

- (1)  $q_{i+1,j}.val = f_{i+1,j} : X_i \rightarrow \Omega$ , где  $i \in [1..(t-2)]$ ,  $j \in 1..|X_{i+1}|$
- (2)  $f_{i+1,j} = T + \sum_t (q_{i,k}.val \sim q_{i,k}.w_k)$ ;  $\forall q_{i,k}: j \in Con$ ; где  $i \in [1..(t-2)]$ ,

$$j \in [1..|X_{i+1}|], k \in [1..|X_{i+1}|];$$

То есть, до того, как мы начинаем вычислять значения датчиков, принадлежащих долям от 2 до  $(t-1)$ , их значение равно  $T$ , а пришедшие с датчиков значения суммируются так: к текущему значению датчика  $q_{(i+1),j}$  прибавляем результат квазисравнения ожидаемого значения со значением датчика, принадлежащего предыдущей доле, связанного через  $Con$  с датчиком  $q_{(i+1),j}$ .

Вектор  $FA$  сопоставляет  $i$ -му датчику высшего порядка процедуру  $r_i$ , которая добавляет в мультимножество  $B$  подмножество действий  $B_i \subseteq A$ , которые связаны с  $i$ -м датчиком высшего порядка элементом матрицы  $S$ , удовлетворяющим зерну  $\xi$ , то есть:

$$(3) \quad B_i = \{a_k \mid S(q_i, a_k) \neq 0; S(q_i, a_k).lv \leq \xi \leq S(q_i, a_k).rv\},$$

$$\forall q_i : val = T \text{ имеем } r_i \in FA : B \cup B_i, \text{ где } i \in [1..|RG^t.M|], k \in [1..|A|].$$

Следует отметить, что разные действия могут иметь один и тот же приоритет и при этом срабатывать для одного и того же вектора входных значений. Варьируя  $FP$ , мы можем по-разному интерпретировать такую ситуацию. Например, для случаев, когда порядок действий с одним и тем же приоритетом содержательно не важен, мы можем, не ограничивая общности, задать ещё одно отношение порядка  $\varphi$  для действий с одинаковым приоритетом, такое что:

$$\forall a_i, a_z \in B : S(q_j, a_i).ord = S(q_j, a_z).ord \text{ имеем } a_i \varphi a_z, \text{ для } i < z, i \in [1..|A|], z \in [1..|A|], j \in [1..|M|] \text{ или}$$

$$\forall a_i, a_z \in B : S(q_j, a_i).ord = S(q_k, a_z).ord \text{ имеем } a_i \varphi a_z, \text{ для } j < k, i \in [1..|A|], z \in [1..|A|], j \in [1..|M|], k \in [1..|M|].$$

Таким образом, мы можем задавать лексикографический порядок по индексам действий, *выходящих* из одного датчика, который вызывает данные действия. Если же действия *выходят* из разных датчиков, то имеет место лексикографический порядок по индексам этих датчиков.

$FP$  может также быть таким, что действия с одинаковым приоритетом будут запускаться параллельно.

Рассмотрим пример использования решающего графа (рис. 3) для задачи диагностики.

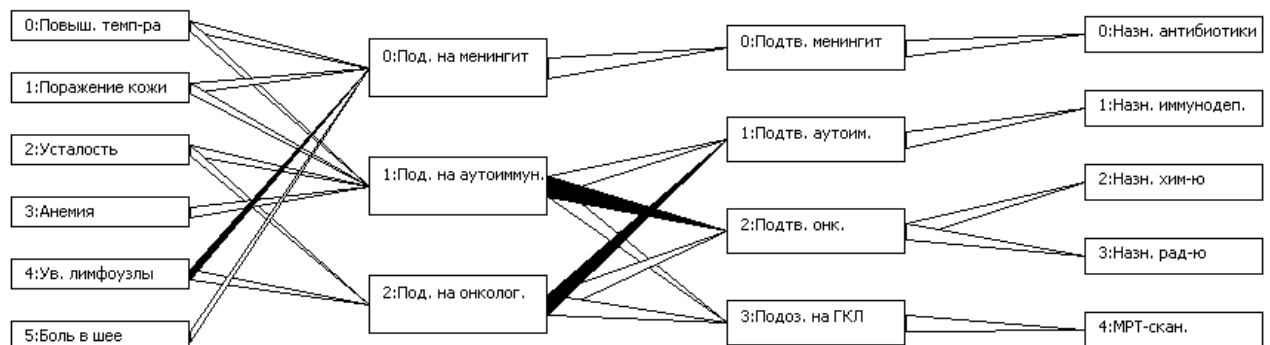


Рис. 3. Пример решающего графа

В данном примере мы имеем набор простых симптомов – повышенная температура, поражение кожи, усталость, анемия, увеличенные лимфоузлы, боль в шее. Показатели этих симптомов (датчиков) мы получаем на входе, значения остальных датчиков вычисляются функцией интерпретации. Так, например, при повышенной температуре, поражении кожи, боли в шее и НЕ увеличенных лимфоузлах мы получаем «истинным» подозрение на менингит, если же показания хоть одного из этих датчиков будут отличаться от ожидаемых, то предположение менингита будет «ложным» или «ошибочным». Увеличенные лимфоузлы и усталость дают нам подозрение на онкологическое заболевание, а набор симптомов из повышенной температуры, поражения кожи, усталости и анемии дают нам «подозрение на аутоиммунное заболевание». В данном примере хорошо виден смысл введения дополнительной промежуточной доли (третьей): когда одновременно возникает подозрение и на аутоиммунное заболевание и на онкологическое – мы получаем подозрение на «новое» заболевание – гистиоцитоз из клеток Лангерганса, т.е. оперировать можно не только «простыми симптомами», но и «комплексными». После постановки диагноза решающий граф предлагает терапию или дополнительное обследование. Так, например, при подтверждённом онкологическом заболевании в графе предписано назначение химио- и радиотерапии.

В этом примере не используется нечёткое определение действий, которое может быть задано матрицей  $S$  решающего графа, и позволяет среди прочего явно указывать корреляцию между действиями.

В третьей главе также определены такие свойства решающего графа как полнота/неполнота, однозначность/неоднозначность, эквивалентность.

Будем говорить, что решающий граф  $RG$  является полным, если для любого вектора входных значений  $(v_h)$  существует непустое подмножество действий  $B$ :

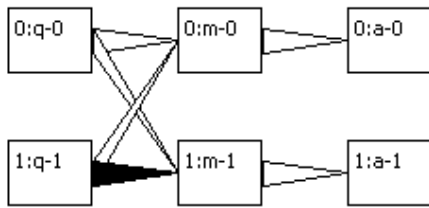
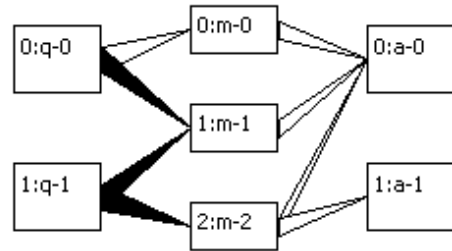
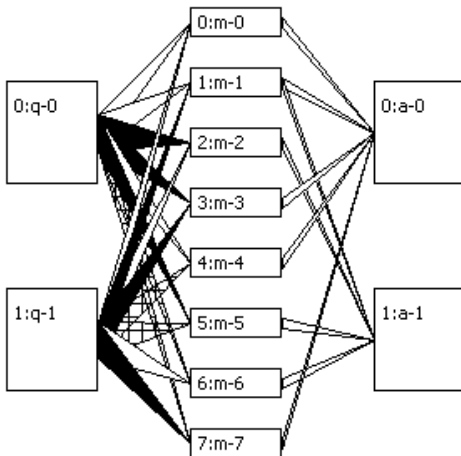
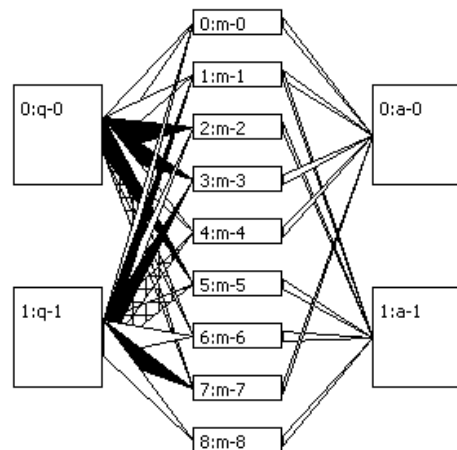
$RG$  – полный  $\Leftrightarrow \forall (v_h)$  функция интерпретации сопоставляет  $B \neq \emptyset$ .

Будем говорить, что решающий граф  $RG$  является неоднозначным, если существует такой вектор  $(v_h)$ , которому функция интерпретации сопоставляет не менее двух датчиков из  $M$  со значением «истина».

$RG$  – неоднозначный  $\Leftrightarrow \exists (v_h): M' \in M$ , где  $M' = \{q_i | val = T\}$ ,  $|M'| \geq 2$ .

Решающий граф может быть:

- неполным и однозначным (рис. 4);
- неполным и неоднозначным (рис. 5);
- полным и однозначным (рис. 6);
- полным и неоднозначным (рис. 7).

Рис. 4. Неполный и однозначный  $RG^3$ Рис. 5. Неполный и не однозначный  $RG^3$ Рис. 6. Полный и однозначный  $RG^3$ Рис. 7. Полный и неоднозначный  $RG^3$ 

Будем говорить, что решающий граф  $RG^t$  эквивалентен решающему графу  $RG^p$ , если:

- $RG^t.X_1 = RG^p.X_1$  – то есть у них идентичный набор датчиков первого порядка;
- $RG^t.A = RG^p.A$  – то есть у них идентичный набор идентификаторов действий;
- любому вектору входных значений ( $v_h$ ) функция интерпретации решающего графа  $RG^t(RG^p)$  ставит в соответствие тот же набор действий, что и для решающего графа  $RG^p(RG^t)$ ;

Доказано утверждение: для любого  $t$ -дольного решающего графа  $RG^t$  ( $t \geq 4$ ) существует эквивалентный трёхдольный решающий граф  $RG^3$ . Приводится алгоритм построения эквивалентного трёхдольного решающего графа. Используя утверждение и алгоритм, можно находить решение за наименьшее время вне зависимости от начальной топологии модели.

Использование решающих графов для визуализации и формализации продукционных правил знаний даёт следующие преимущества:

- удобство визуализации и наглядность процесса принятия решений;
- возможность описывать различные компоненты системы на разном уровне абстракции;
- возможность легко строить и модифицировать уже существующие модели;

- возможность распараллеливать различные этапы принятия решений;
- возможность автоматически преобразовывать модель в более эффективную (по скорости принятия решений) без изменения системы правил (на основе утверждения о сведении моделей) и др.

**В четвёртой главе** приводится описание программных средств, разработанных в рамках диссертационного исследования. Представлена их архитектура, используемые модели данных и связь этих данных с математической моделью, представленной в главе 3.

Целью программных разработок – получение комплекса для дискретно-событийного ИМ с помощью решающих графов, содержащего инструменты для поддержки всех этапов имитационного эксперимента.

На основе математических алгоритмов описанных в главе 3 разработаны библиотеки, содержащие функции верификации, интерпретации, сводимости и обработки моделей. Данные библиотеки легли в основу следующего программного комплекса *G-IPS Ultimate*, поддерживающего такие режимы работы как:

- создание и редактирование моделей, представленных решающими графами;
- тестирование и прояснение внутренней логики модели (режим кларификации);
- режим тестовой имитации с остановкой после каждого шага имитации;
- режим рабочей имитации с демонстрацией динамики процесса;
- режим демонстрации ранее проведённой имитации.

Комплекс *G-IPS Ultimate* реализован как *Windows*-приложение (рис. 8) и содержит ряд инструментов, позволяющих строить модели для самых разных задач, например:

- средства для обучения оперативно-диспетчерского персонала сложных систем (тренажёры);
- экономические модели, основанные на методе Монте-Карло;
- экспертные системы;
- игровые модели;
- модели алгоритмов (например, для выявления «узких» мест алгоритма, или для поиска «узлов» алгоритма, которые можно реализовать более эффективно), и др.

Приложение *G-IPS Ultimate* имеет следующие объёмные характеристики.

Среда разработки	Число строк исходного кода	Объём исходного кода	Объём машинного кода
<i>CodeGear RAD STUDIO 2007</i>	> 10200	> 230 Кб	~2.3 Мб

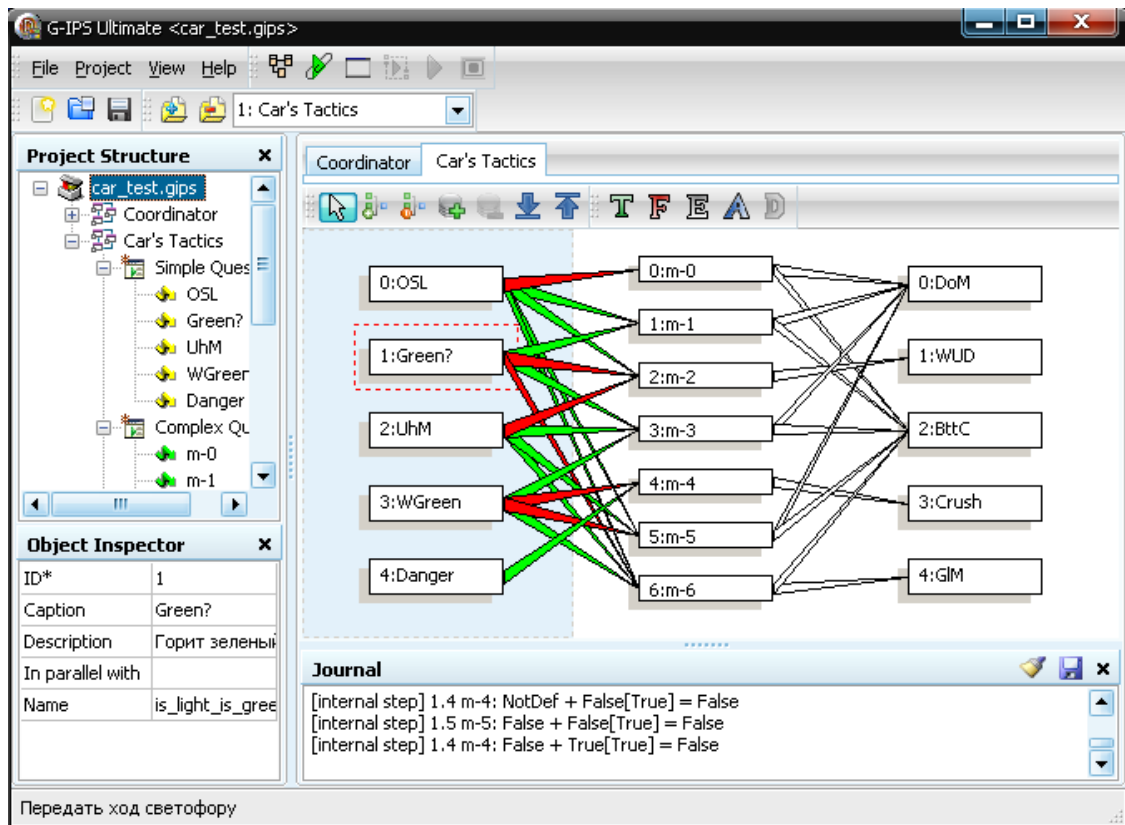


Рис. 8. Окно редактора G-IPS Ultimate

Для обеспечения взаимодействия комплекса *G-IPS Ultimate* с внешними модулями были разработаны специальные низкоуровневые интерфейсы:

- интерфейс для подключения внешних *dll*-модулей, реализующих взаимодействие конкретной модели с внешней средой;
- интерфейс для подключения *dll*-модулей, расширяющих функциональное наполнение комплекса *G-IPS Ultimate*, например, генераторов, валидаторов моделей и т.п.

*G-IPS Ultimate* обеспечивает взаимодействие построенной модели с внешним модулем (*DLL*), имитирующим внешнюю среду для системы. Шаг прогона модели можно условно разделить на два этапа. На первом этапе (рис. 9) *G-IPS Ultimate* запрашивает значения элементарных датчиков из подключаемого внешнего модуля (*DLL*), используя низкоуровневый интерфейс *API GipsInclude*. Помимо самого значения функции, из *dll*-модуля может поступить команда редактору *G-IPS Ultimate* через функцию обратного вызова, с помощью которой реализован ряд команд, таких как «добавить запись в текстовый журнал событий», «приостановить выполнение», «сохранить какую-либо величину в памяти» и др.

После того как показатели простых датчиков получены, редактор вычисляет значения остальных датчиков и составляет список действий, подлежащих выполнению. Затем наступает второй этап прогона модели – выполнение предписанных действий (рис. 10).



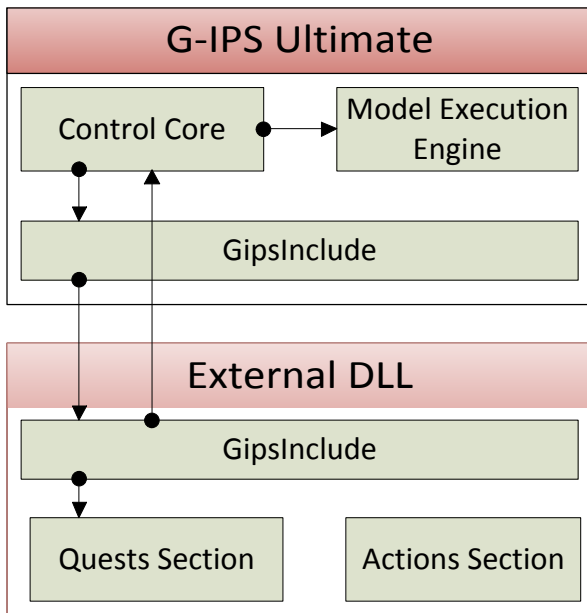


Рис. 9. Схема опроса условий *G-IPS Ultimate* через *API GipsInclude*

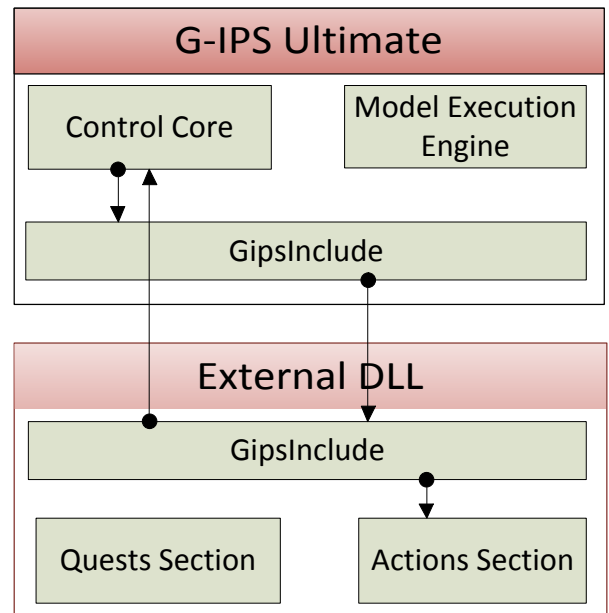


Рис. 10. Схема выполнения действий через *API GipsInclude*

Аналогичным образом реализован *API GipsHostPlugins*, используемый для подключения таких плагинов, расширяющих функциональное наполнение комплекса *G-IPS Ultimate*. Данные плагины могут поддерживать различные этапы моделирования – это могут быть генераторы моделей, валидаторы, проверяющие как корректность модели, так и различные её свойства (например, полнота или неоднозначность), различные сборщики статистики и многое другое.

В отличие от многих рассмотренных в главе 2 программных средств данный комплекс содержит ряд функциональных возможностей помогающих построить корректную (с точки зрения логики) модель: это журнал, в который заносятся все промежуточные шаги имитации, это режим кларификации, позволяющий визуально наблюдать логику принятия решения и режим пошаговой имитации. *G-IPS Ultimate* имеет возможность расширения функционального наполнения за счёт подключения плагинов (валидаторов, генераторов, анализаторов прогона и др.), представленных динамическими библиотеками (*dll*). Данный комплекс поддерживает возможность подключения программных модулей, задающих внешнюю среду для конкретной модели, что позволяет не только узнать результат имитации, но и визуализировать протекание имитационного эксперимента. Он может быть использован в учебном процессе при выполнении лабораторного практикума в таких дисциплинах как: экспертные системы, теория игр, теория принятия решений, интеллектуальные системы, а так же при выполнении курсовых и дипломных проектов.

Для демонстрации работы разработанного программного средства в данной главе подробно описаны следующие модели.

1. Модель (5 решающих графов) алгоритма решения *NP*-трудной задачи поиска индекса структурной спектральной сложности графа в базисе цепей заданной длины.
2. Модель (6 графов) решения игровой ситуации на примере игры «*RacMan*».
3. Модель (5 графов), имитирующая бизнес-процесс функционирования приёмного отделения медицинского центра.

С помощью специальных плагинов, таких как: генератор конструктивного перечисления семейства решающих графов, пакетный преобразователь произвольных моделей к трёхдольному виду и пакетный обработчик моделей была подтверждена полнота и корректность работы комплекса *G-IPS Ultimate*, получены оценки эффективности преобразования моделей к трёхдольному виду. Тестирование проводилось на семействе четырёхдольных решающих графов, содержащих по 2-3 вершины в каждой доле (299943 пары решающих графов, проверка более сложных случаев продолжается).

**В приложениях** приведены автоматически сгенерированные шаблоны исходного кода для демонстрационных примеров, список сокращений и исходный код комплекса *G-IPS Ultimate*.

## ОСНОВНЫЕ ВЫВОДЫ И РЕЗУЛЬТАТЫ

1. Исследованы модели представления знаний, представления времени и подходы к имитационному моделированию (дискретно-событийный, системно-динамический и агентный). Проанализированы достоинства, недостатки и области применения рассмотренных моделей, представлений и подходов. Показана значимость разработок в области дискретно-событийного моделирования.

2. Выполнен сравнительный анализ более 20 наиболее распространённых инструментов имитационного моделирования с целью определения места разрабатываемого программного средства среди уже существующих. Представлены такие характеристики продуктов, как стоимость, платформа, возможности, предметная ориентированность, функциональные возможности и ограничения. Также рассмотрены программные средства имитационного моделирования, используемые в учебном процессе в российской высшей школе. На основе проведённого анализа выявлены достоинства и недостатки существующих средств имитационного моделирования.

3. Предложена оригинальная модель представления экспертных знаний в виде системы решающих графов на основе продукционных правил. Доказаны её основные свойства, используемые при последующей

программной реализации (полнота, неоднозначность, сводимость). Основной целью являлся поиск компромисса: простота создания и понимания человеком с одной стороны и мощность модели с другой стороны. Также предложена визуализация модели в структурной форме, удобной для восприятия человеком.

4. На основе предложенной модели разработаны оригинальные программные средства, поддерживающие все необходимые этапы имитационного эксперимента: построение модели, верификация модели, прогон (имитация), визуализация, анализ результатов имитации. Дополнительно созданы средства накопления и импорта/экспорта моделей.

5. Создан оригинальный программный комплекс *G-IPS Ultimate*, поддерживающий визуальное проектирование имитационных моделей в виде решающих графов. Ядро комплекса *G-IPS Ultimate* реализовано в среде *CodeGear Rad Studio 2007*, объём авторского исходного кода составляет более 10200 строк авторского кода объёмом более 230 Кб, размер скомпилированного модуля – 2,3 Мб.

6. В работе приведены примеры разработок с использованием комплекса *G-IPS Ultimate*. Тестирование комплекса проводилось как с помощью объёмных вычислительных экспериментов на конструктивно перечисленных классах решающих графов (299943 пары решающих графов), так и на синтетических и реальных примерах: реализации алгоритма решения *NP*-трудной задачи поиска топологических индексов графов в базисах цепей, реализации игры «*PacMan*», имитации принятия решений в работе приёмного отделения медицинского центра и др.

7. К настоящему моменту разработанный программный комплекс внедрён ФГБУ «ФНКЦ ДГОИ им. Дмитрия Рогачева» Министерства здравоохранения РФ для моделирования бизнес-процессов, а также в учебный процесс РГСУ.

**Список работ, опубликованных по теме диссертации:****Работы, опубликованные в перечне ведущих рецензируемых научных журналов и изданий, рекомендованных ВАК:**

1. **Карпухин И. Н., Кораблин Ю. П., Незнанов А. А.** Модели времени в имитационном моделировании / Научно-Техническая Информация. Серия 2. Информационные процессы и системы, № 5, 2012. – С. 16-21.
2. **Карпухин И. Н., Кораблин Ю. П., Незнанов А. А.** Программное средство для имитационного моделирования с помощью графовых моделей / Программные продукты и системы, № 2 (98), 2012. – С. 156-158.

**В других изданиях:**

3. **Карпухин И. Н., Незнанов А. А.** Системы имитационного моделирования учебного назначения в российской высшей школе / Современные проблемы информационной безопасности и программной инженерии, 2011. – С. 132-142.
4. **Карпухин И. Н.** Программное средство учебного назначения для имитационного моделирования с помощью графовых моделей / Всероссийский конкурс научно-исследовательских работ студентов и аспирантов в области информатики и информационных технологий, Т. III, 2012. – С. 218-222.
5. **Karpukhin I. N., Korablin Yu. P., Neznanov A. A.** The Models of Time in Simulation / Automatic Documentation and Mathematical Linguistics, № 3, 2012.
6. **Карпухин И.Н., Незнанов А.А.** Программные средства имитационного моделирования процессов принятия решений реального времени // Научные доклады НПК «Интегрированные модели, мягкие вычисления, вероятностные системы и комплексы программ в искусственном интеллекте» (ИММВИИ-2009), Т.2. – М.: Физматлит, 2009. – С. 132-140.
7. **Карпухин И. Н., Яркин С. В., Кохов В. А.** Генераторы средних по сложности структур для исследования базовых алгоритмов структурной информатики // Тринадцатая ежегодная международная научно-техническая конференция студентов и аспирантов «РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА», Т.1. – М.: МЭИ (ТУ), 2007. – С. 367-368.