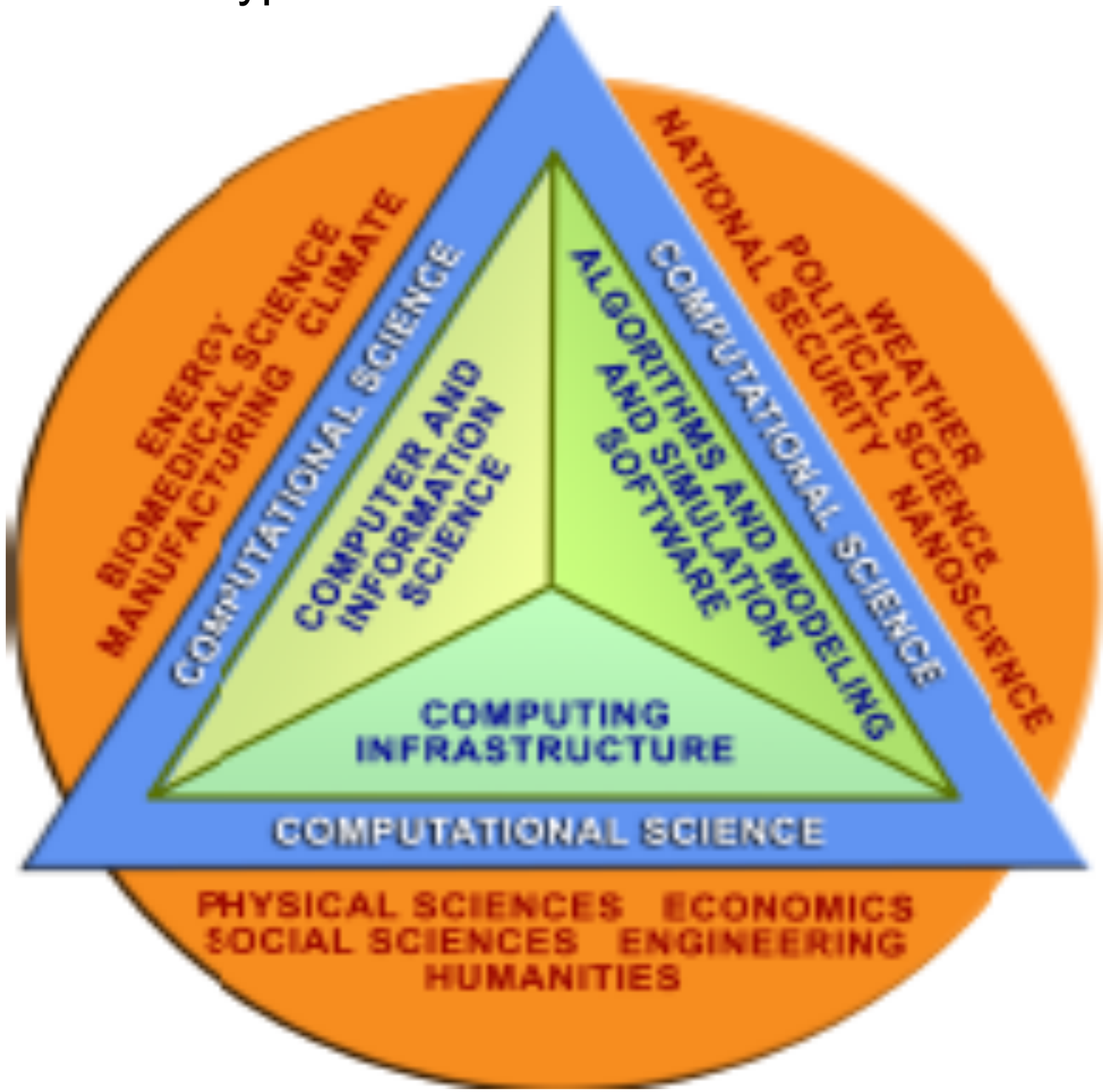
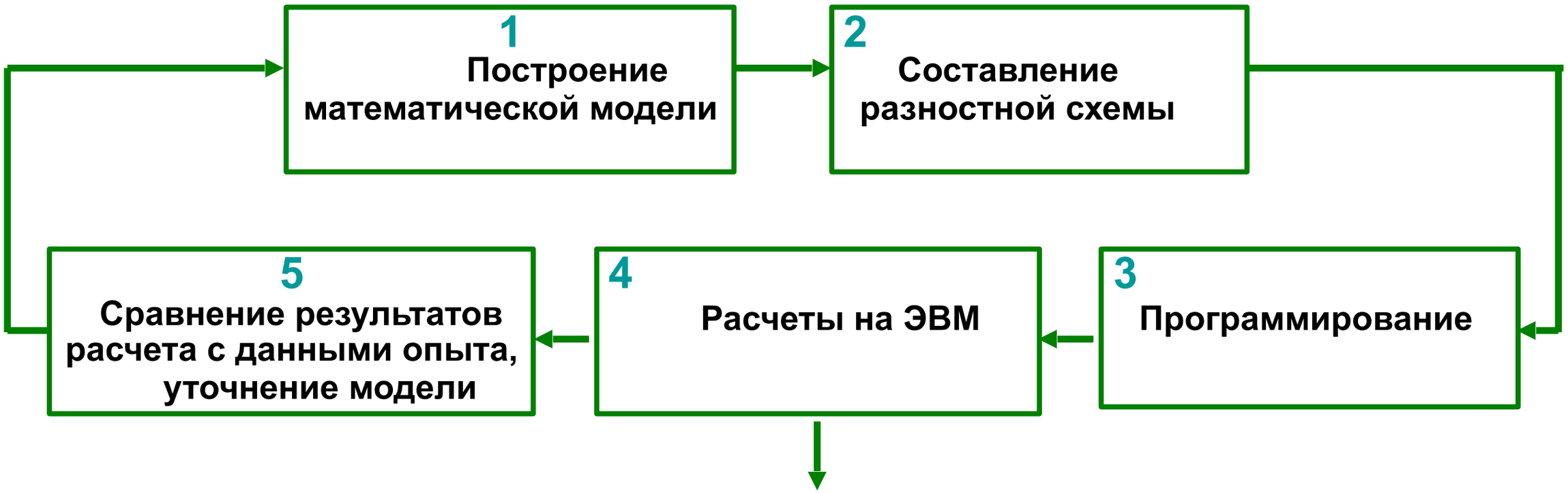
Современные инструментальные средства автоматизации моделирования

Доктор технических наук, доцент Михайлов В.В.

COMPUTATIONAL SCIENCE: Ensuring America's Competitiveness
(доклад PITAC – President's Information Technology Committee, 2005)
– новая стратегическая компьютерная инициатива для обеспечения конкурентоспособности США



А.А.Самарский – математическое моделирование и вычислительный эксперимент



1986 г. – Постановление ЦК КПСС и СМ СССР об усилении научно-исследовательских работ в области математики. Предусматривались создание службы математического моделирования и разработка общегосударственной программы широкого использования математического моделирования в различных отраслях народного хозяйства.

Полезность методов исследования операций в повседневной научной работе

Методы	Относительная ценность
Теория вероятностей (и статистические оценки)	0,182
Экономический анализ (оценки эффективности затрат)	0,150
Имитационное моделирование	0,143
Линейное программирование	0,120
Управление запасами	0,097
Массовое обслуживание (теория очередей)	0,085
Сетевые модели (упорядочение операций)	0,072
Модели замены	0,042
Теория игр	0,040
Динамическое программирование	0,031
Методы поиска	0,020

Методы, наиболее часто используемые во внутрифирменном планировании

Методы	Частота использо-вания	Процент
Имитационное моделирование	60	29
Линейное программирование	43	21
Сетевые методы (включая ПЕРТ и МКП)	28	14
Теория управления запасами	24	12
Нелинейное программирование	16	8
Динамическое программирование	8	4
Целочисленное программирование	7	3
Теория массового обслуживания	7	3
Прочие	12	6

Организационные проблемы комплексного моделирования (проблемы финансирования, менеджмент)

- **Высокая стоимость программного обеспечения КМ**

На настоящий момент среди систем имитационного моделирования ведущее положение на мировом рынке занимают следующие программные продукты: *Arena* (Rockwell Automation, Inc.), *ProModel* (ProModel Corp.), *AutoMod* (Applied Materials, Inc.), *eM-Plant* (Tecnomatix Technologies, Ltd, Германия), *ReThink* (Gensym Corp.), *Extend* (Imagine That, Inc.), *AnyLogic* (ООО «Экс Джей Текнолоджис», РФ), *Simul8* (Simul8 Corp.), *SIMPROCESS* (CACI Products Company), *FlexSim* (Flexsim Software Products, Inc).

- **Относительно высокая стоимость проектов КМ**
- **Высокая стоимость оплаты труда специалистов в области КМ**
- **Ошибки управления проектов, связанных с КМ**

Актуальность и особенности имитационного моделирования сложных объектов

КОНЦЕПЦИЯ	СОВРЕМЕННОЕ СОСТОЯНИЕ	САМОРЕГУЛИРУЮЩИЕСЯ ВЫЧИСЛЕНИЯ
Самоконфигурация	*Корпоративные центры обработки данных используют оборудование и платформы различных производителей. Установка, конфигурирование и интеграция систем требуют больших затрат времени и изобилуют ошибками.*	*Автоматическое конфигурирование компонентов и систем в соответствии с высокоуровневыми правилами. Остальная часть системы настраивается автоматически и бесконфликтно.*
Самооптимизация	*Система имеет сотни устанавливаемых вручную, нелинейно меняющихся параметров, и с каждой новой реализацией их число растет.*	*Компоненты и системы постоянно ищут возможность увеличить свою производительность и эффективность.*
Самовосстановление	*Выявление проблем в крупных, сложных системах требует работы целой группы программистов в течение нескольких недель.*	*Система автоматически выявляет, диагностирует и исправляет локализованные программные и аппаратные проблемы.*
Самозащита	*Определение и восстановление после атак или серии порождающих друг друга ошибок вручную.*	*Система автоматически защищается от вредоносных атак или порождающих друг друга ошибок. Она использует средства раннего предупреждения для прогнозирования и предупреждения общесистемных сбоев.*

GPSS

- GPSS – это неординарное явление м мире программирования конца -60-х/ начала 70-х годов. Это был один из самых удачных на то время проблемно-ориентированных языков программирования. Первая версия языка GPSS была введена в IBM в октябре 1961 г. Проблемной областью GPSS являются системы массового обслуживания (системы с очередями). Основой имитационных алгоритмов в GPSS является дискретно-событийный подход, разработанный Джеффри Гордоном. В GPSS разработчикам удалось четко пройти по грани как соответствия проблемной области (по терминологии, функциям, методологии исследования и т. д.), так и эффективности программирования. GPSS по праву считается языком, который сделал моделирование практической деятельностью, доступной широким массам программистов
- **www:gpss.ru**
- **Рыжиков Ю.И. Имитационное моделирование: теория и прктика. Алтекс-А, 2004. 380 с.**
- **Бражников А.Н. Имитационное моделирование. Возможности GPSS WORLD. Реноме, 2006. 439 с.**
- **Варжапетян А.Г. Имитационное моделирование на GPSS/H. ГУАП. 2008.**
- **Шрайдер Г.Д. Моделирование с использованием GPSS. Эл.версия.**

GPSS

- Основные показатели качества обслуживания:
- • общее количество обслуженных заявок за какой- либо промежуток
- времени;
- • пропускная способность – среднее число заявок, обслуженных в
- единицу времени;
- • доля заявок обслуженных;
- • доля заявок, получивших отказ;
- • время пребывания заявки в системе (от момента поступления заявки
- в систему до момента завершения ее обслуживания);
- • среднее время обслуживания (функция распределения времени
- обслуживания);
- • средняя длина очереди;
- • среднее время ожидания;
- • загрузка каналов – коэффициент использования (как доля времени, в
- течение которого ОУ было занято) – характеризует степень простоя ОУ;

GPSS

- В языке GPSS реализована блочно-ориентированная концепция
- структуризации моделируемого процесса, разработанная с ориентацией
- на описание систем массового обслуживания (СМО).
- Структура моделируемого процесса изображается в виде потока,
- проходящего через обслуживающие устройства (ОУ), очереди, ключи и другие элементы СМО.
- Модель имеет блочную структуру. Моделируемый процесс представляется как поток заявок в системе обслуживания. Блоки интерпретируются как ОУ.
- Заявки (транзакты) конкурируют между собой за место в ОУ, образуют очереди перед ОУ, если они заняты.
- Дуги на блок-схеме – потенциальные потоки заявок между ОУ. Существуют истоки и стоки этих заявок. В этом случае блок-схема модели описывает маршруты движения заявок в системе.

GPSS

- GPSS относится к классу **процессно- (транзактно) - ориентированных**
- **систем моделирования**. GPSS является способом алгоритмизации дискрет'
- ных динамических систем. Примеры моделируемых объектов: транспорт'
- ные объекты, склады, производственные системы, магазины, торговые
- объекты, сети ЭВМ, системы передачи сообщений. Функциональная структура GPSS рассматривается на двух уровнях.
- • **1 уровень** определяется комбинацией основных функциональных
- объектов таких, как:
- устройства;
- памяти;
- ключи (логические переключатели);
- очереди;
- транзакты.
- • **2 уровень** – блок-схема модели, составленная из типовых блоков,
- между которыми перемещаются транзакты.

Аппаратно ориентированные объекты GPSS:

- • **Транзакты** являются абстрактными подвижными элементами,
- которые являются аналогами различных объектов реального мира
- (сообщения, транспортные средства, люди, детали т.д.) Это
- динамические функциональные элементы GPSS, которые отражают
- реальные заявки на обслуживание.
- Транзакты двигаются по модели, появляются в ней с той же
- интенсивностью, что и реальные заявки.
- • **Устройства** моделируют объекты, в которых может происходить
- обработка транзактов, что связано с затратами времени. Устройства
- являются аналогами каналов СМО (каждое устройство в данный
- момент времени может быть занять лишь одним транзактом).

• **Памяти** предназначены для моделирования объектов, обладающих емкостью.

• **Логические переключатели** принимают значение включено/выключено, позволяют изменять пути следования транзактов в модели.

Аппаратно ориентированные объекты GPSS:

- Статистические объекты: очереди, таблицы
- На 2 уровне модель на языке GPSS имеет
- наглядное графическое представление в виде **блок-схемы**.
- **Блоки – операционные объекты GPSS**. Каждый блок имеет
- стандартное обозначение. Последовательность блоков – это есть
- последовательность операторов на языке GPSS. Любую модель на языке
- GPSS можно представить в виде совокупности блоков, между которыми
- перемещаются транзакты, они имеют вход-выход, в блоках реализуются
- все действия, связанные с обслуживанием транзакта (создание и
- уничтожение транзактов, изменение параметров транзакта, управление
- потоками транзактов, и т.д.). Таким образом, на языке GPSS составляется и реализуется функциональная блок-схема.

MatLab

- Развивающийся пакет с широким спектром приложений, за счет использования пакетов ToolBox - матричный анализ, обработка сигналов и изображений, анализ динамических систем, задачи матфизики, оптимизация, работу с картографическими изображениями, нейронные сети, генетические алгоритмы, нечеткая логика и др.
- Справочная система с описанием встроенных функций, доступ из командной строки к кратким сведениям о функциях, электронные книги по MatLab и ToolBox.
- Открытость кодов
- Интеграция с Microsoft Word и Excel, доступ к базам данных Access, SQL, Oracle.
- Символьные вычисления с использованием ядра пакета Maple
- Визуальная среда GUIDE для написания приложений с графическим интерфейсом пользователя
- Программный интерфейс API для связи среды MatLab с программами, написанными на языке C или Fortran.

Литература

- Дьяконов В.П. MatLab. СПб.: Питер, 2001. 553 с.
- Ануфриев И. Самоучитель ManLab 5.3/6.x СПб.:БХВ-Петербург, 2002. 709 с.I
- Гультяев А. Визуальное моделирование в среде MatLab.СПб.: Питер, 2000. 430 с.
- Потемкин И.Г. Система инженерных и научных расчетов MatLab 5.x : в 2-х т. М.:ДИАЛОГ-МИФИ, 1999.
- www.exponenta.ru

Maple

- Пакет символьных вычислений и численных решений
- Алгебраические преобразования
- Интегрирование и дифференцирование аналитических ыфункций в общем виде
- Решение систем дифференциальных уравнений в аналитическом виде
- Матричные и векторные вычисления
- Решение задач из теории графов и теории групп
- Комбинаторика и теория групп
- Визуализация результатов вычислений с использованием 2D и 3D графики
- Статистический анализ
- Более 20 библиотек по различным областям знаний
- Древовидная справочная система с примерами, которые можно проигрывать со своими данными

Справочная информация

- Прохоров Г.В., Леденев М.А., Колбеев В.В. Пакет символьных вычислений Maple V. М.: Петит, 1997. 198 с.
- www.exponenta.ru

Model-Vision

- Поддержка гибридных моделей
- Описание дискретных объектов картами состояний UML
- Описание непрерывных объектов системами уравнений в математической нотации
- Поддержка компонентного моделирования с направленными и ненаправленными связями
- Инкрементный транслятор и диагностика ошибок
- Типовая конструкция MV – обобщенный гибридный автомат, как последовательная композиция непрерывных компонент, взаимодействующих через начальные условия

- Колесов Ю.Б., Сениченков Ю.Б. Моделирование систем. Динамические и гибридные системы. СПб.: БХЗ-Петербург, 2006.
- Колесов Ю.Б. Объектно-ориентированное моделирование сложных динамических систем. СПб.: СПбГПУ, 2004. 239 с.
- Колесов Ю.Б., Сениченков Ю.Б. Визуальное моделирование сложных динамических систем.

Формализм гибридного автомата.

Гибридный автомат: H = {t, G, V, C, P, A, F},

где: G – ориентированный граф, вершины которого это множество дискретных состояний автомата, S – дуги графа, E - возможные переходы из состояния в состояние, S0 – начальное состояние, t – непрерывное время

V = (Vc, Vd) – множество непрерывных Vc и дискретных Vd переменных,

C – множество непрерывных отображений, P – множество логических предикатов, A – множество мгновенных действий,

F = {Fc, Fp, Fa} - Fc: C → S – отображение, сопоставляющее множеству непрерывных отображений состояния автомата, Fp: P → E – отображение, сопоставляющее множеству предикатов множество переходов, Fa: A → E – отображение, сопоставляющее множеству мгновенных действий множество переходов

Формы математических моделей, которые непосредственно поддерживаются численными методами.

Система ДУ первого порядка, разрешенная относительно производных:

$$\frac{dx}{dt} = F(x, w.t\},$$

где – F непрерывная функция в окрестности t_0.

Система алгебраических уравнений
0 = G(y, w, t), G - корректна по числу уравнений и структурно не вырождена.

$$A(x)\frac{dx}{dt} = F(x, w, t),$$

A – невырожденная матрица.

$$G(\frac{dx}{dt}, x, w, t) = 0$$

- система АУ относительно производных корректна.

Система дифференциально-алгебраических уравнений индекса 1 в форме Хессенберга.

$$\frac{dx}{dt} = F(x, y, w, t), \quad 0 = G(x, y, w, t).$$

Система дифференциально-алгебраических уравнений индекса 2 в форме Хессенберга.

$$\frac{dx}{dt} = F(x, y, w, t), \quad 0 = G(x, w, t).$$

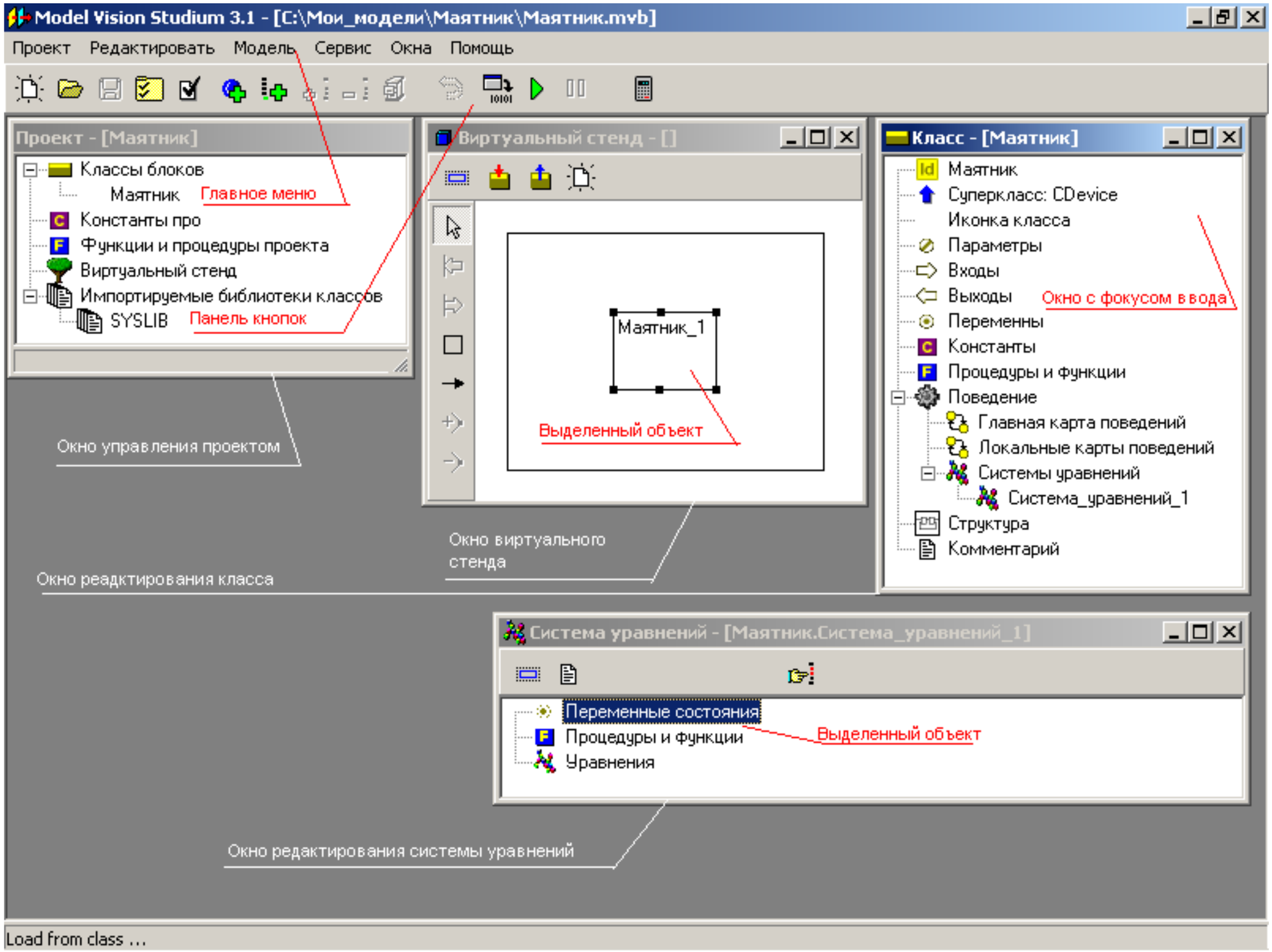
Более удобным является задание непрерывной системы в естественной форме:

$$F(\frac{d^k x}{dt^k}, ... \frac{dx}{dt}, x, y, w., t).$$

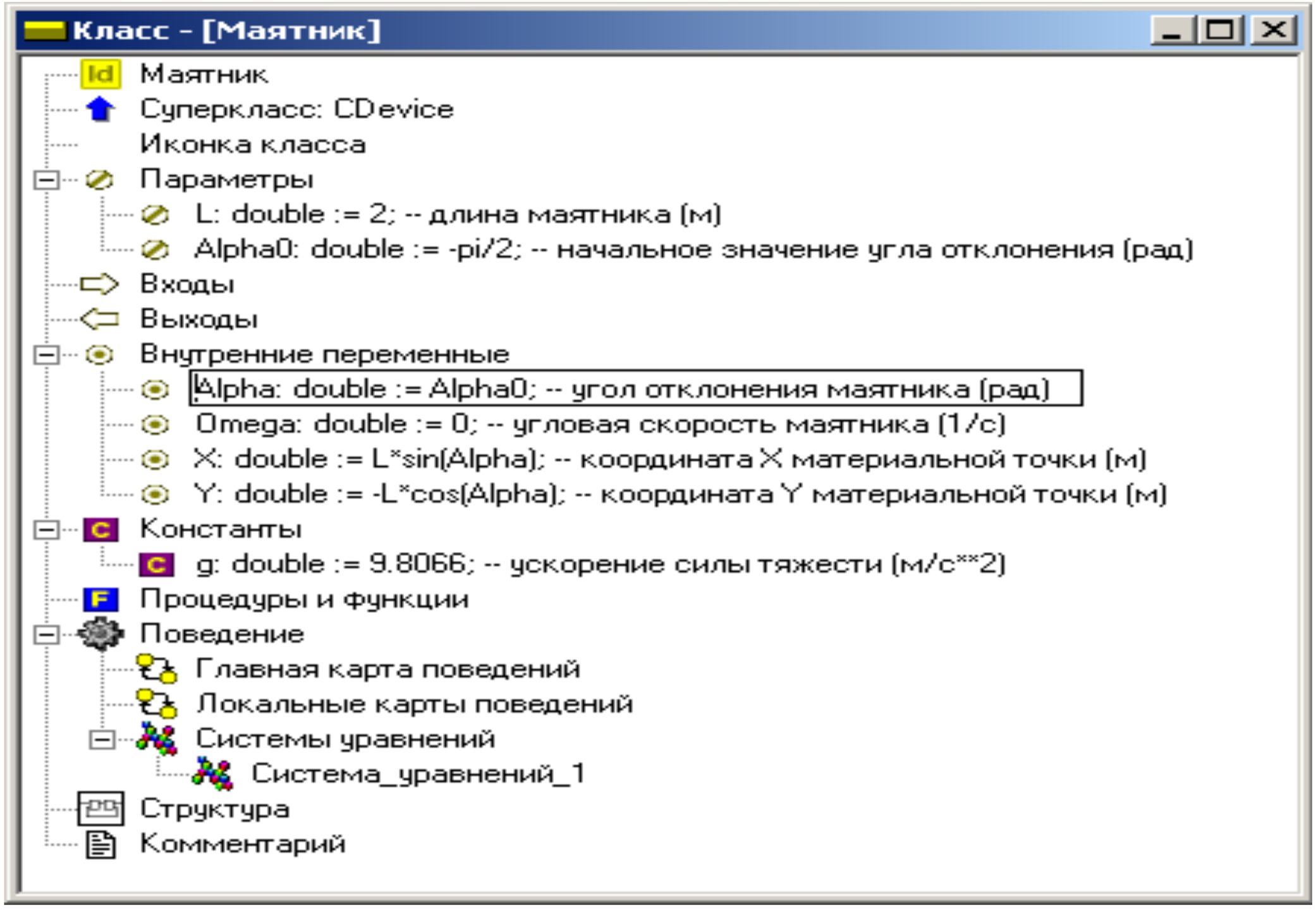
или

$$Q : Fj(\frac{d^k x}{dt^k}, ..., \frac{dx}{dt}, X, Y, t) = 0, \quad j = \overline{1, N}.$$

Главное окно приложений MVS



Окно модели


Класс - [Маятник]
Маятник
Суперкласс: CDevice
Иконка класса
Параметры
L: double := 2; -- длина маятника (м)
Alpha0: double := -pi/2; -- начальное значение угла отклонения (рад)
Входы
Выходы
Внутренние переменные
Alpha: double := Alpha0; -- угол отклонения маятника (рад)
Omega: double := 0; -- угловая скорость маятника (1/с)
X: double := L*sin(Alpha); -- координата X материальной точки (м)
Y: double := -L*cos(Alpha); -- координата Y материальной точки (м)
Константы
g: double := 9.8066; -- ускорение силы тяжести (м/с**2)
Процедуры и функции
Поведение
Главная карта поведений
Локальные карты поведений
Системы уравнений
Система_уравнений_1
Структура
Комментарий

Ввод формул

Редактор Формул - [Маятник . Система_уравнений_1]

Файл Правка Вставка Вид Помощь

-- уравнения углового движения маятника

$$\frac{d\,\mathrm{Alpha}}{d\,t} = \mathrm{Omega}$$

$$\frac{d\,\mathrm{Omega}}{d\,t} = -g\cdot\frac{\sin(\mathrm{Alpha})}{L}$$

$X = L\cdot\sin(\mathrm{Alpha})$ -- вычисление координат

$Y = -L\cdot\cos(\mathrm{Alpha})$

Готово

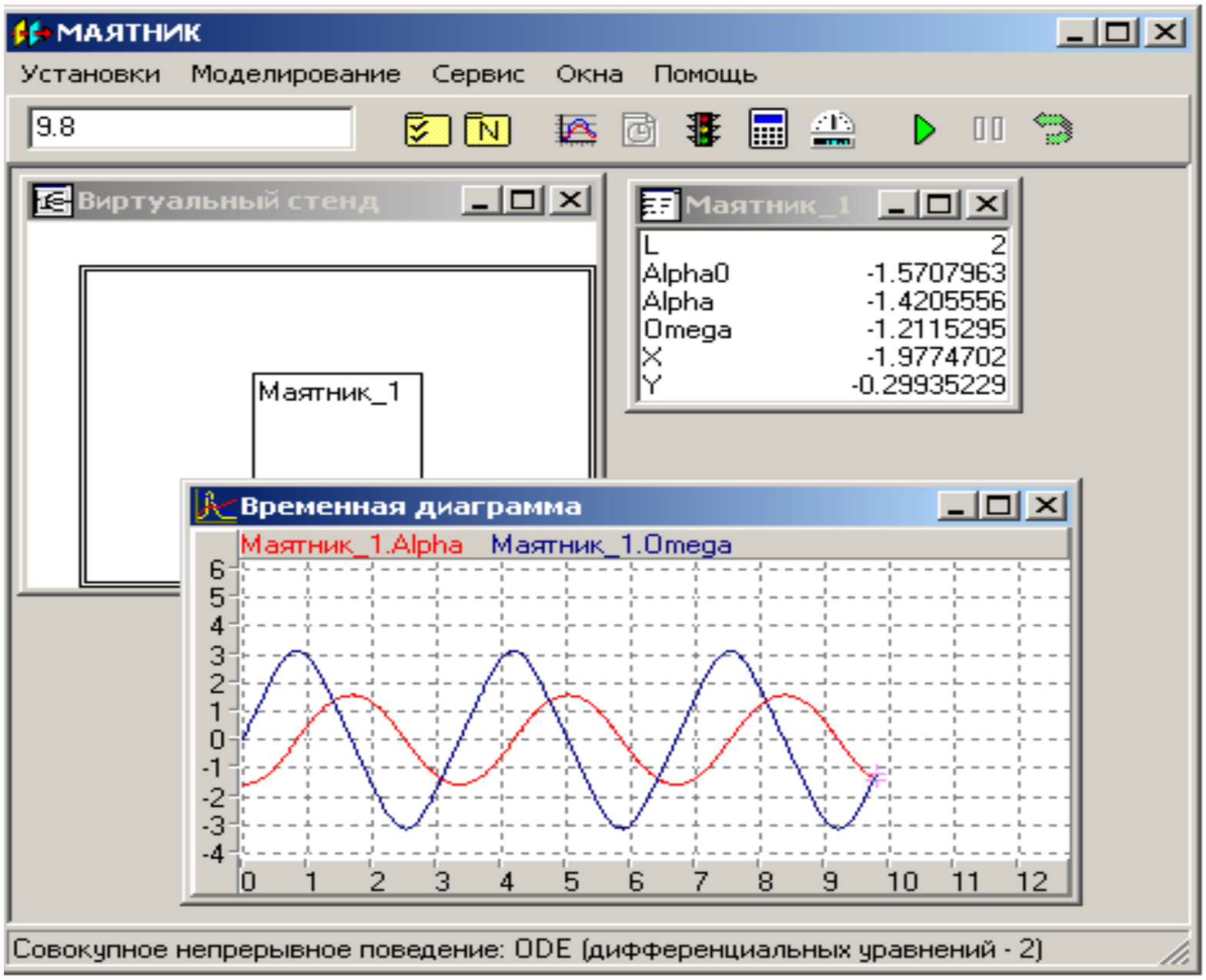
$$\frac{d^2\,\mathrm{Alpha}}{d\,t^2} = -g\cdot\frac{\sin(\mathrm{Alpha})}{L} - mu\cdot\frac{d\,\mathrm{Alpha}}{d\,t}$$

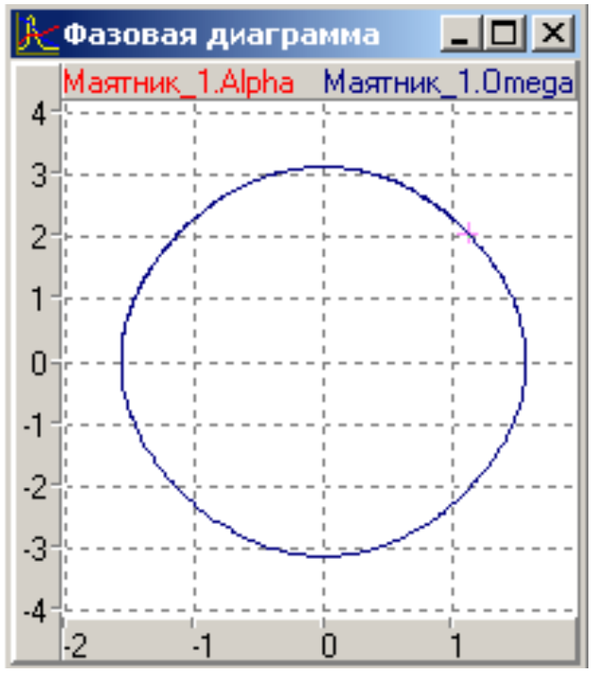
$X = L\cdot\sin(\mathrm{Alpha})$

$Y = -L\cdot\cos(\mathrm{Alpha})$

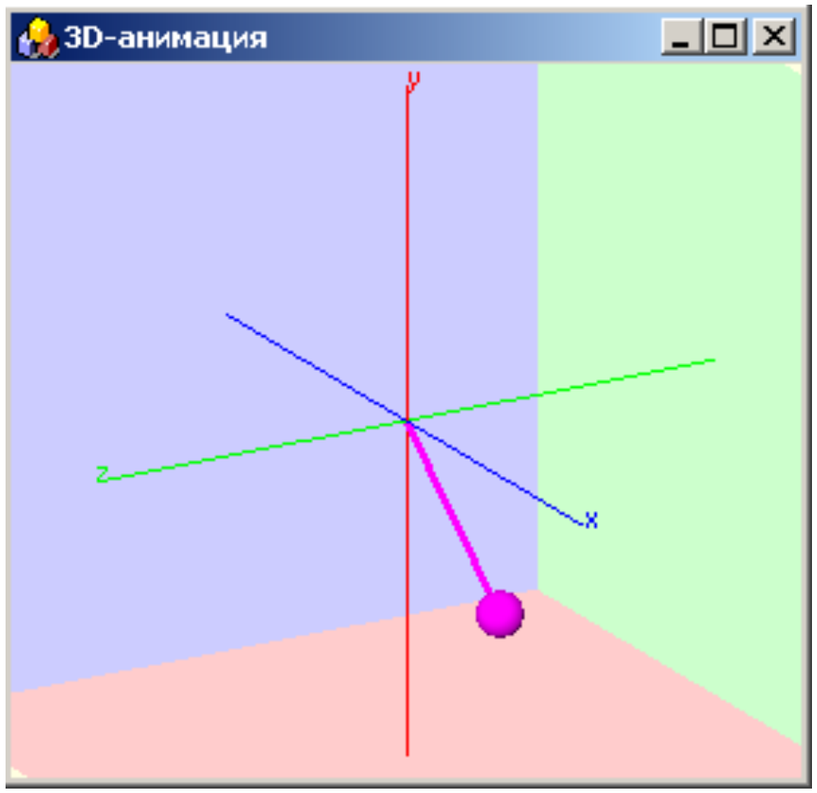
find Alpha,X,Y

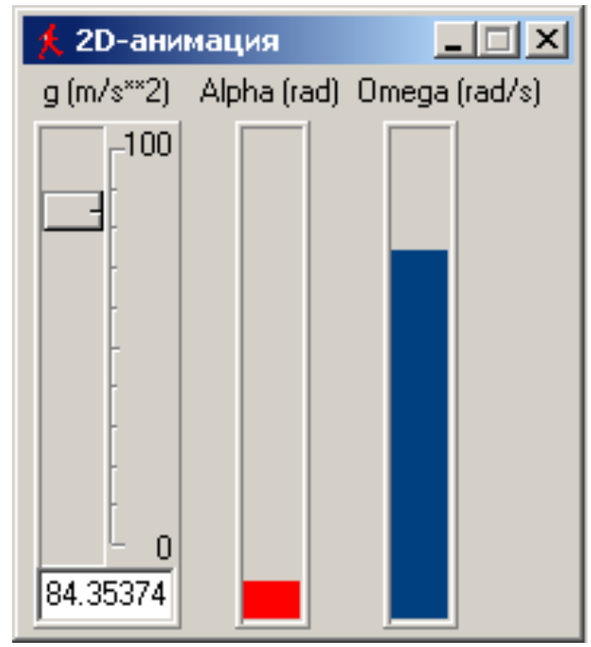
Временные и фазовые диаграммы

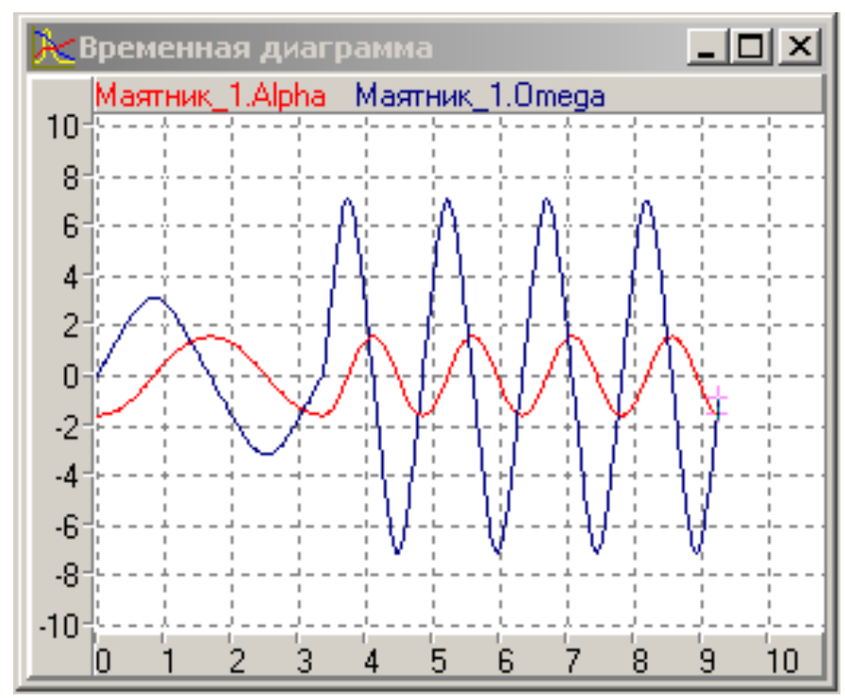




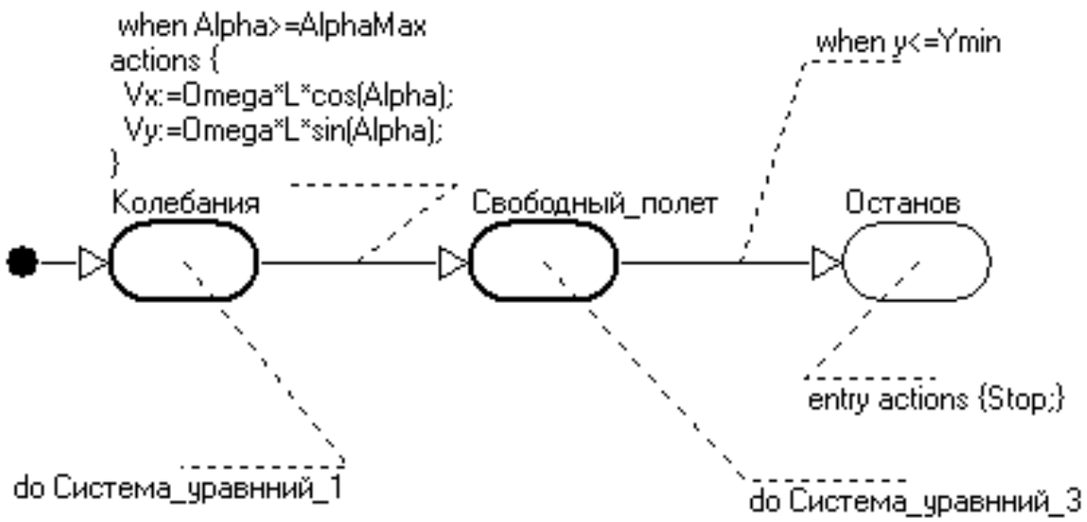
3-D и 2-D анимация, интерактивное вмешательство

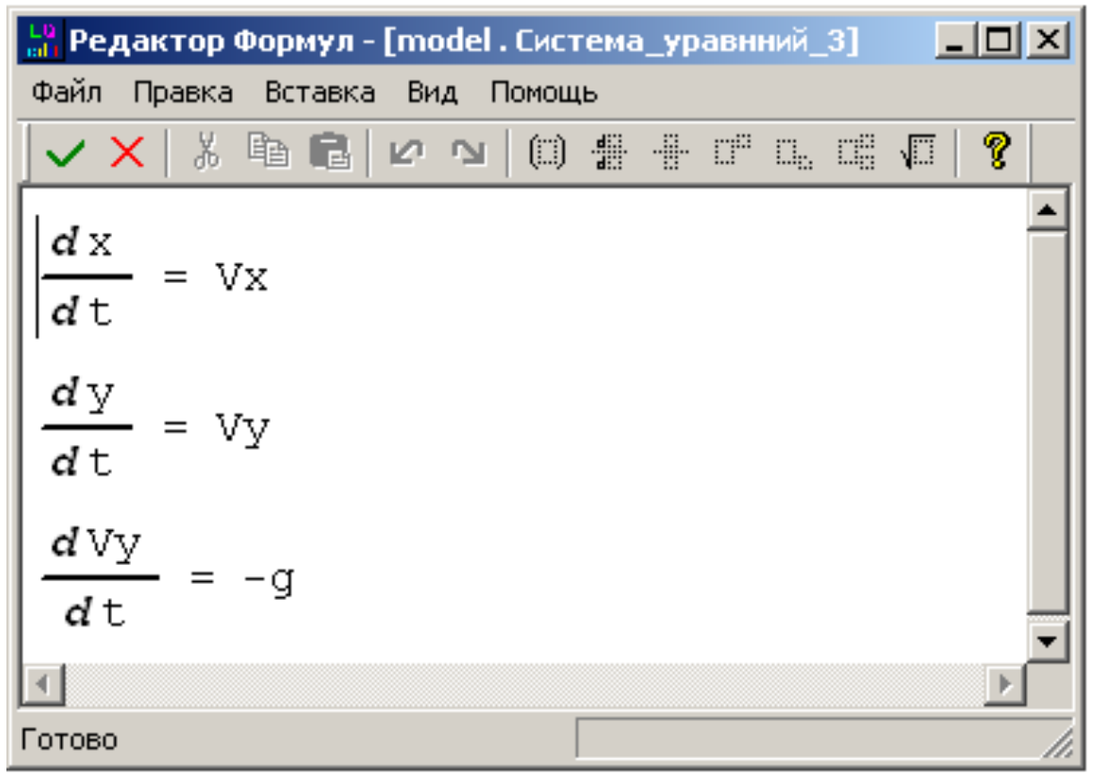


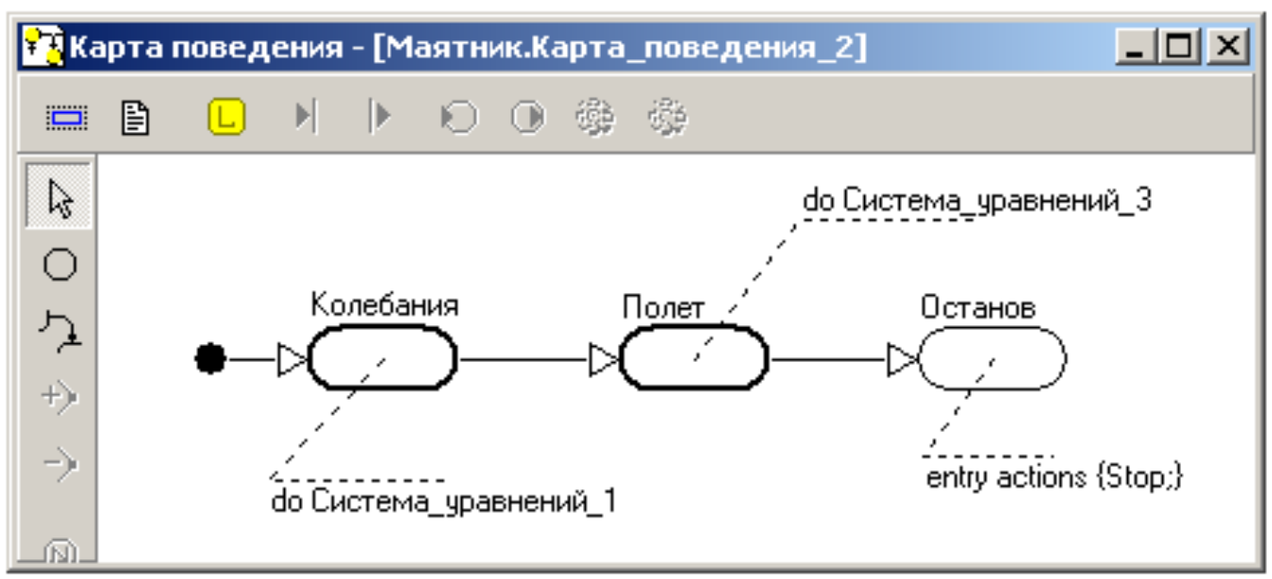




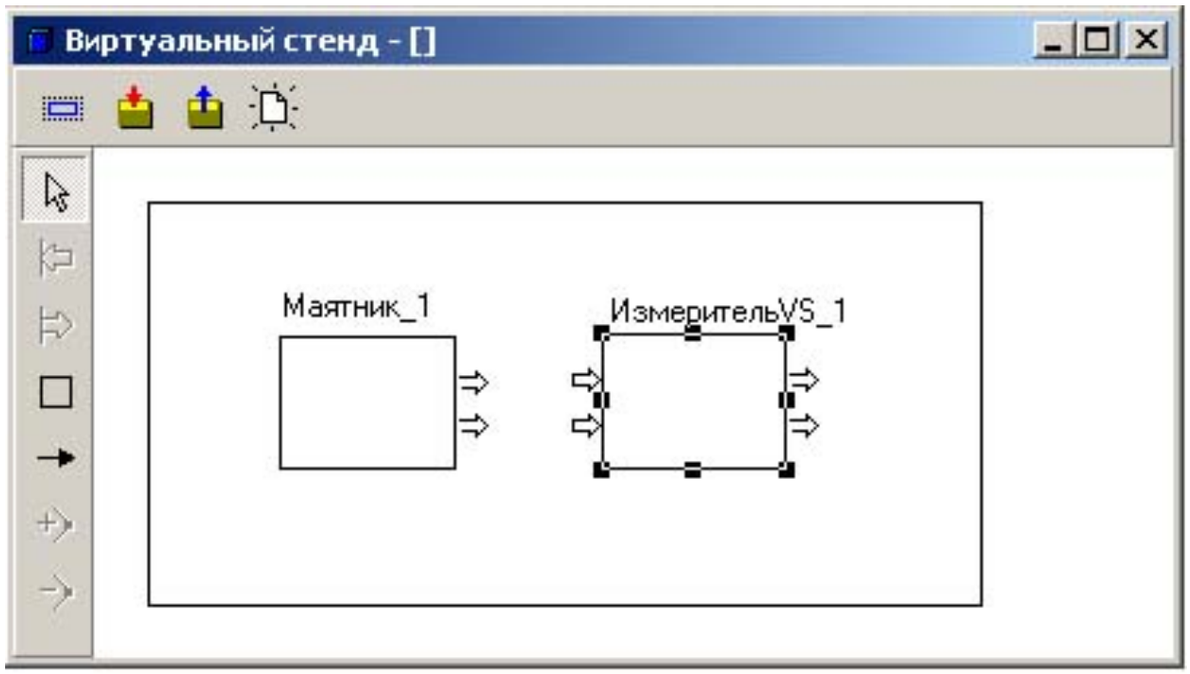
Гибридная модель отрывающегося маятника

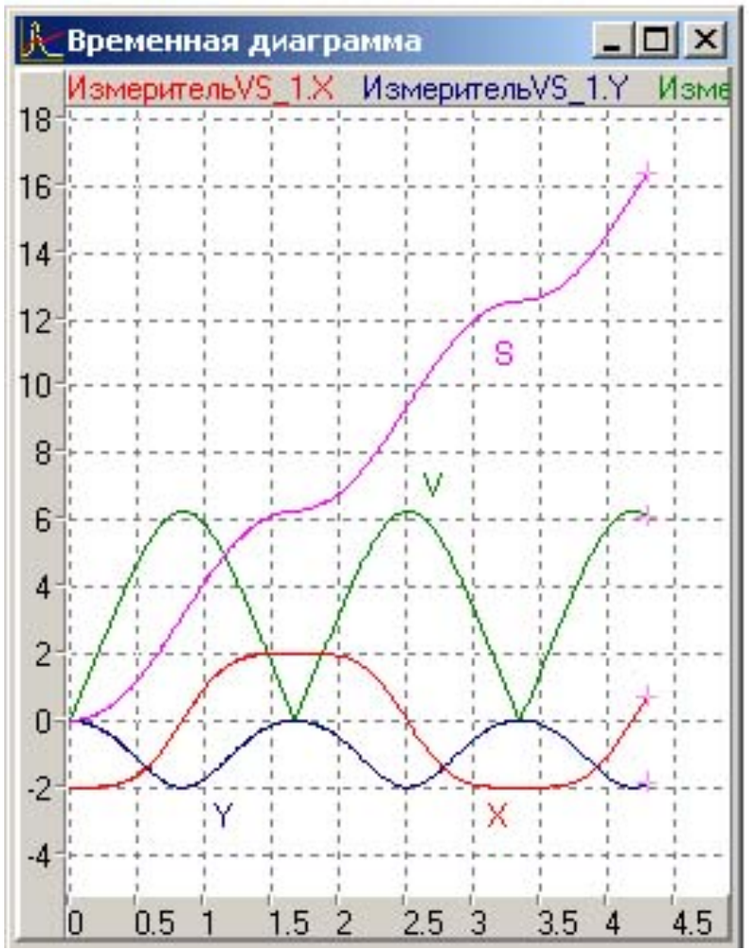






Компонентное моделирование маятника

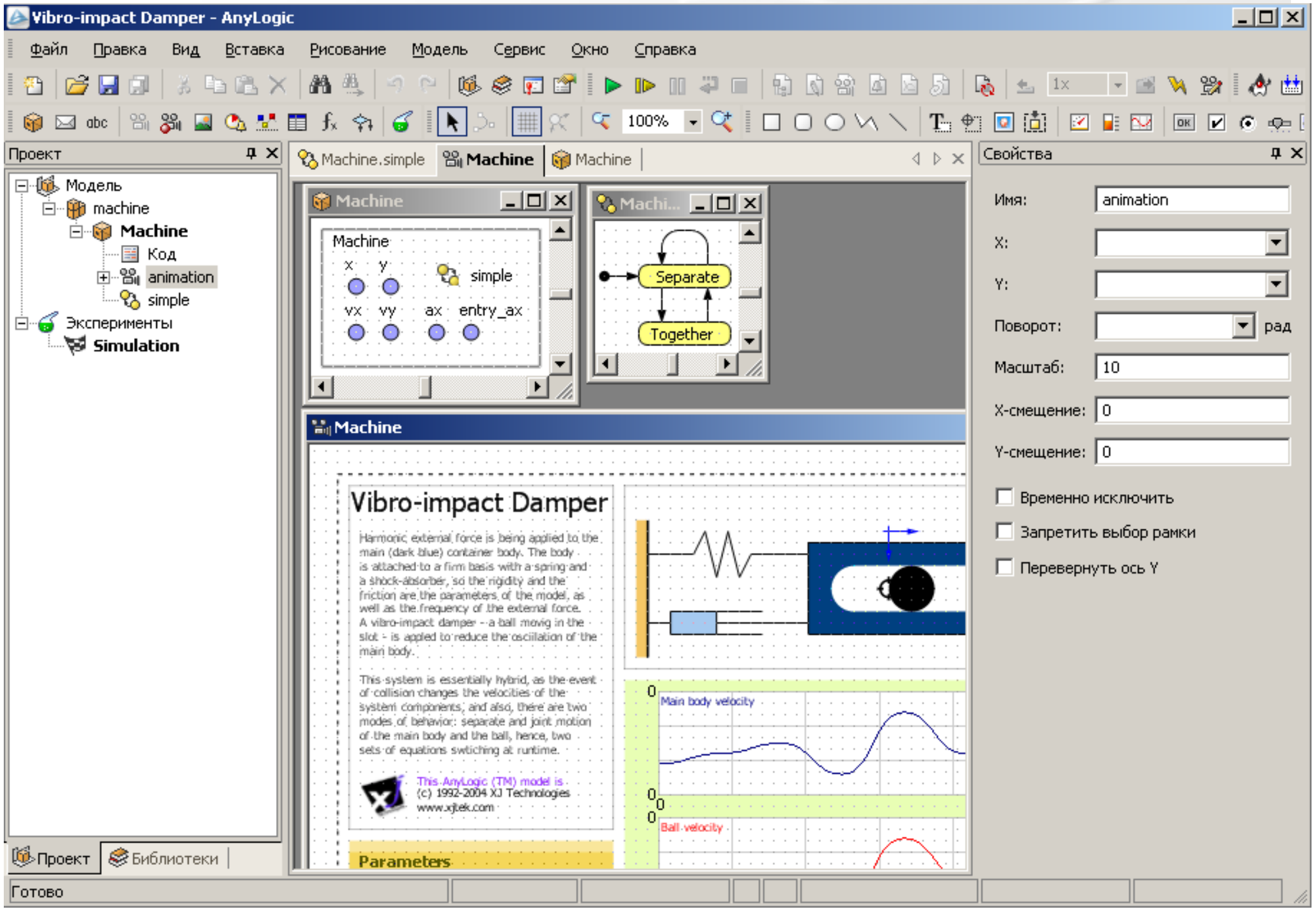




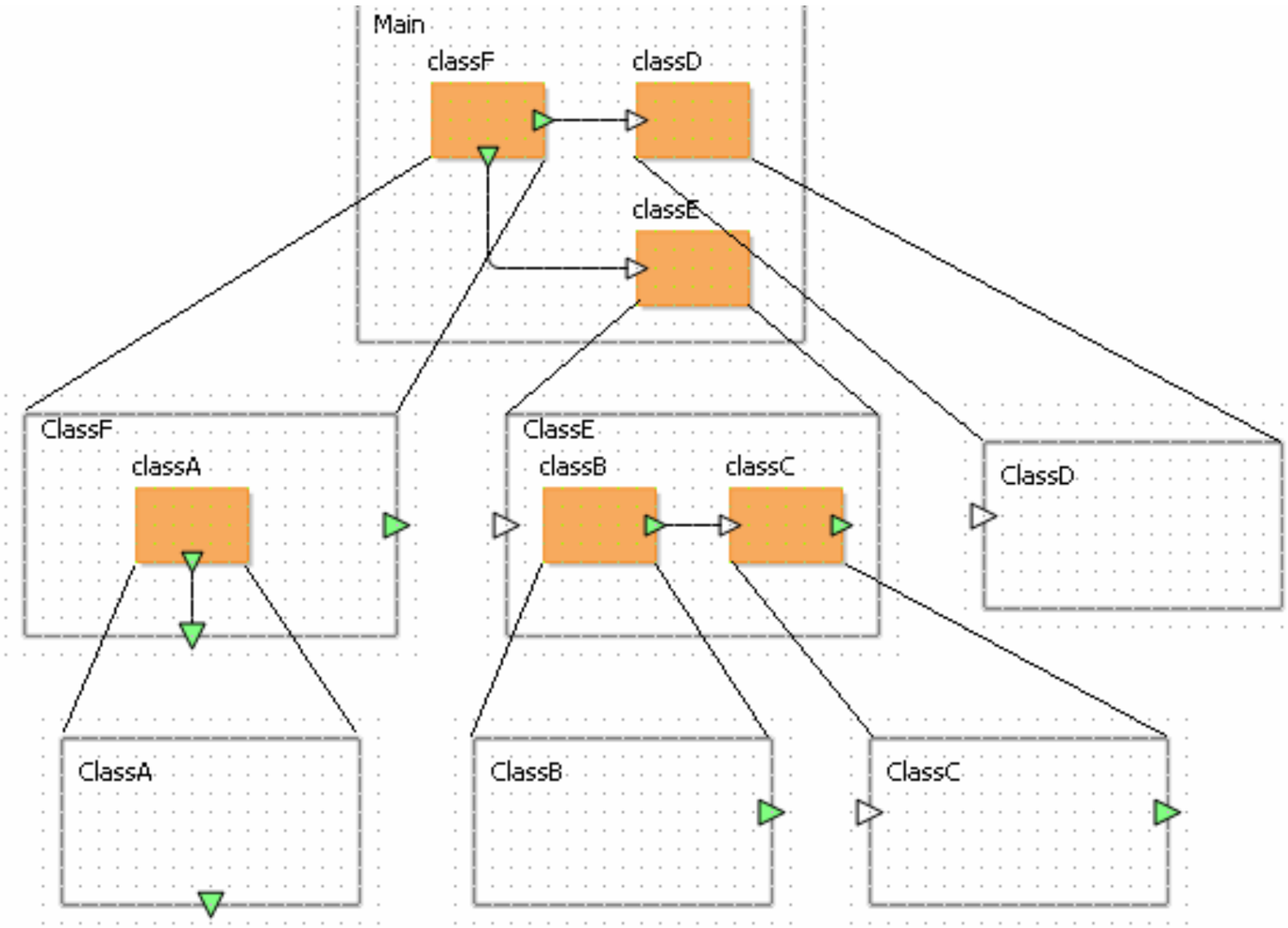
AnyLogic

- AL - это инструментальное средство нового поколения для разработки и исследования имитационных моделей.
- Программный пакет основан на объектно-ориентированном подходе – лучшем на сегодняшний день методе управления сложностью информации.
- Модель – набор взаимодействующих, параллельно функционирующих активностей – активных объектов (АО), которые могут включать любое количество других АО, могут динамически порождаться и исчезать.
- Графическая среда поддерживает проектирование, разработку , документирование модели, проведение компьютерного эксперимента. Визуальное проектирование как принцип разработки моделей.
- Пакет позволяет разрабатывать модели в соответствии основными парадигмами моделирования – динамические системы, системная динамика, дискретно-событийное моделирование, мультиагентные системы.
- AL – надстройка над языком Java, фрагменты кода которого используются при разработке компонент моделей.

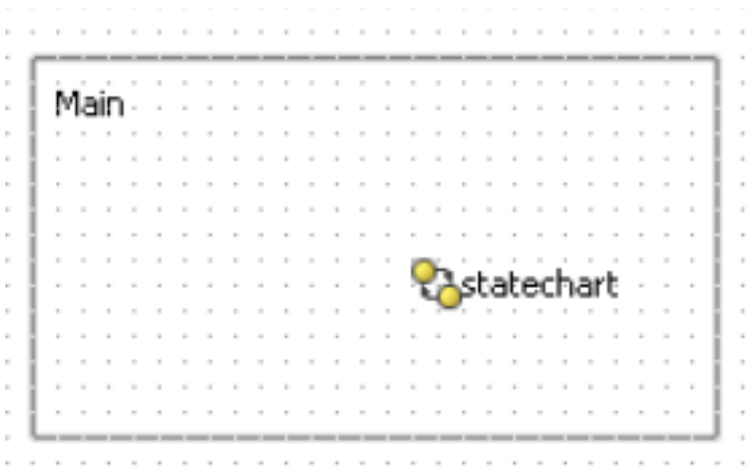
Окно модели

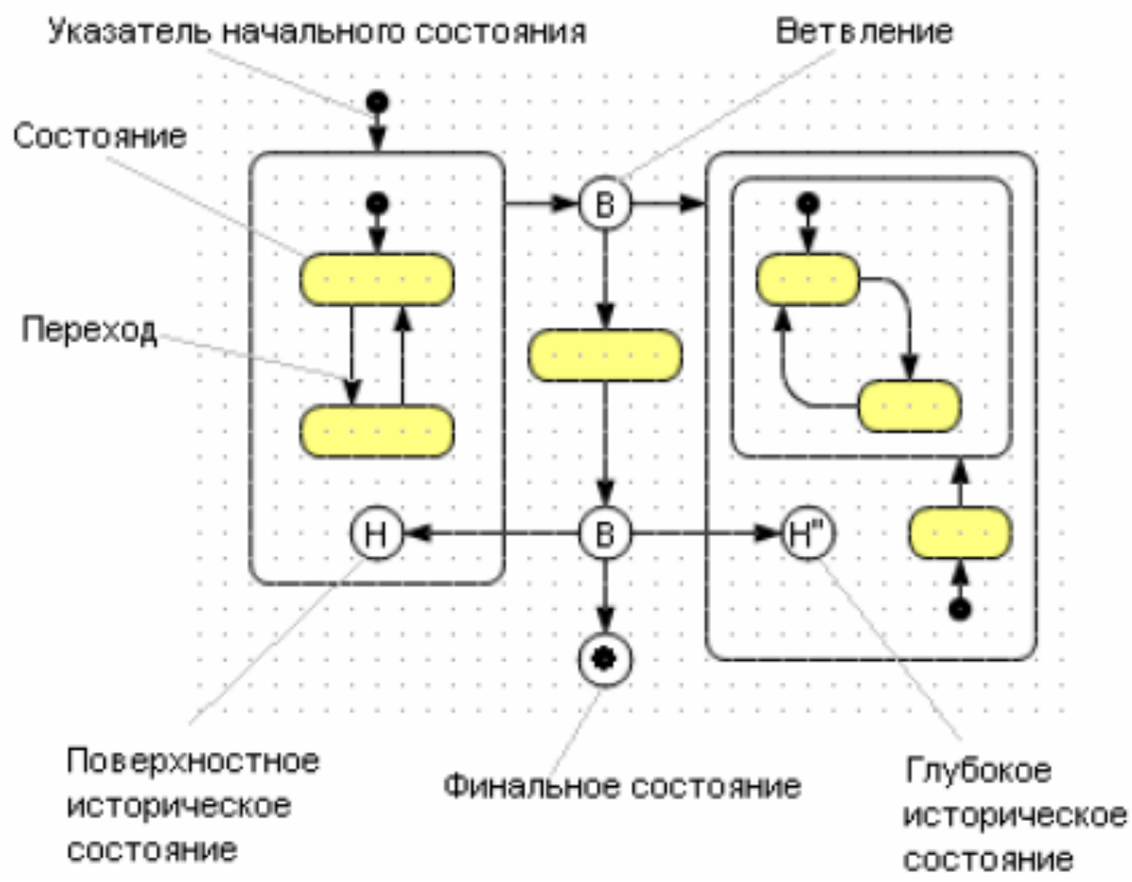

Vibro-impact Damper - AnyLogic
Файл Правка Вид Вставка Рисование Модель Сервис Окно Справка
Проект
Модель
machine
Machine
Код
animation
simple
Эксперименты
Simulation
Machine.simple
Machine
Machine
x y
simple
vx vy ax entry_ax
Separate
Together
Vibro-impact Damper
Harmonic external force is being applied to the main (dark blue) container body. The body is attached to a firm basis with a spring and a shock-absorber, so the rigidity and the friction are the parameters of the model, as well as the frequency of the external force. A vibro-impact damper - a ball moving in the slot - is applied to reduce the oscillation of the main body.
This system is essentially hybrid, as the event of collision changes the velocities of the system components, and also, there are two modes of behavior: separate and joint motion of the main body and the ball, hence, two sets of equations switching at runtime.
This AnyLogic (TM) model is (c) 1992-2004 XJ Technologies www.xjtek.com
Main body velocity
Ball velocity
Parameters
Свойства
Имя: animation
X:
Y:
Поворот: рад
Масштаб: 10
X-смещение: 0
Y-смещение: 0
Временно исключить
Запретить выбор рамки
Перевернуть ось Y
Проект
Библиотеки
Готово

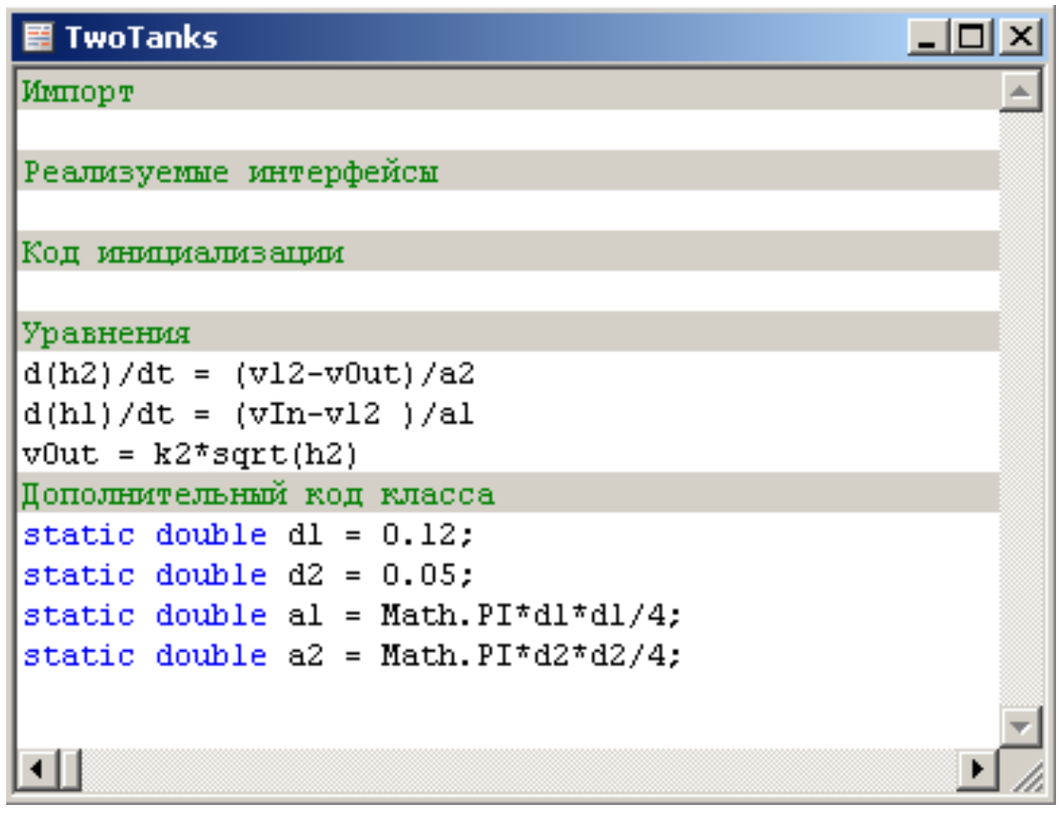
Структура классов AnyLogic



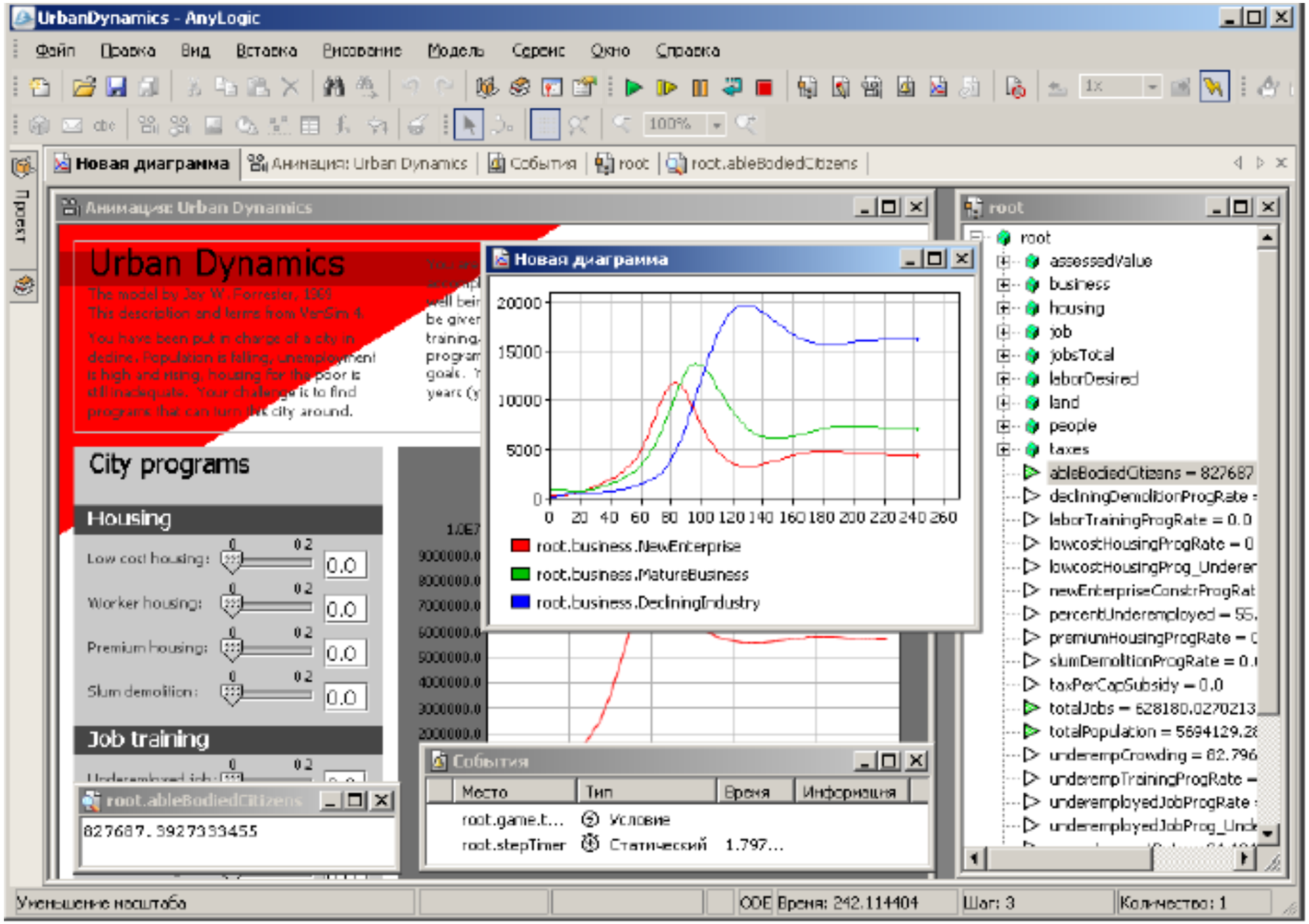
Таймер, стейтчарт, запись уравнений



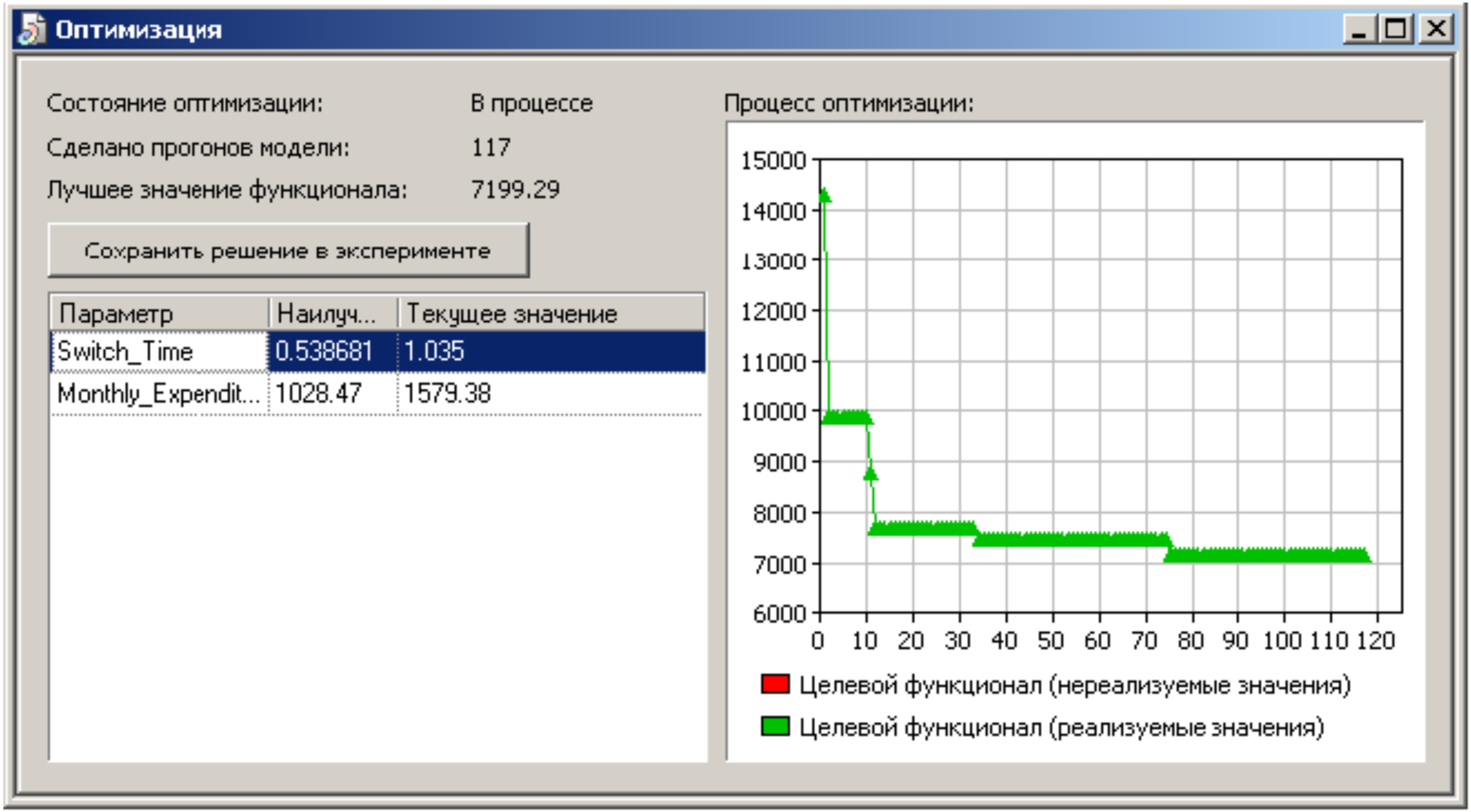




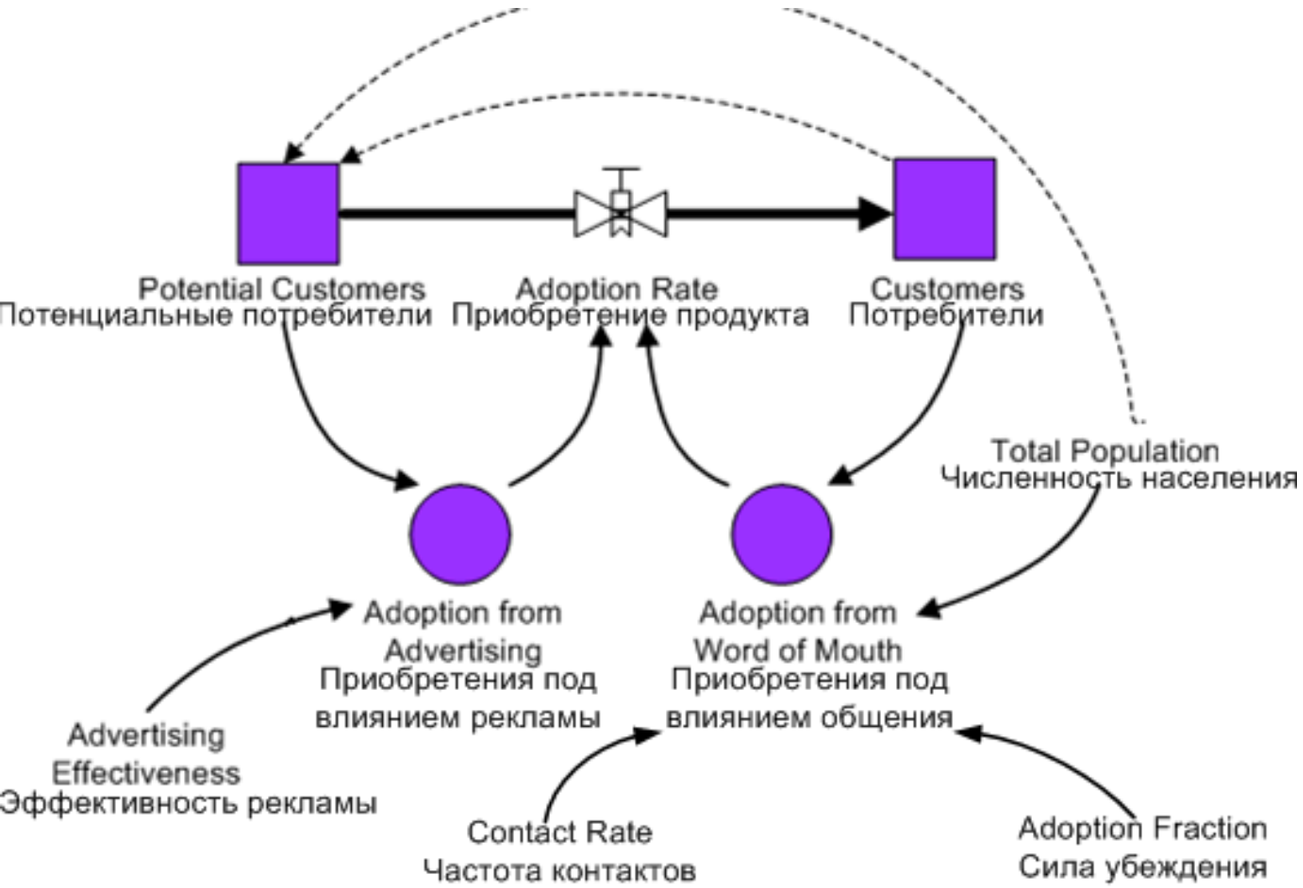
Вывод графиков

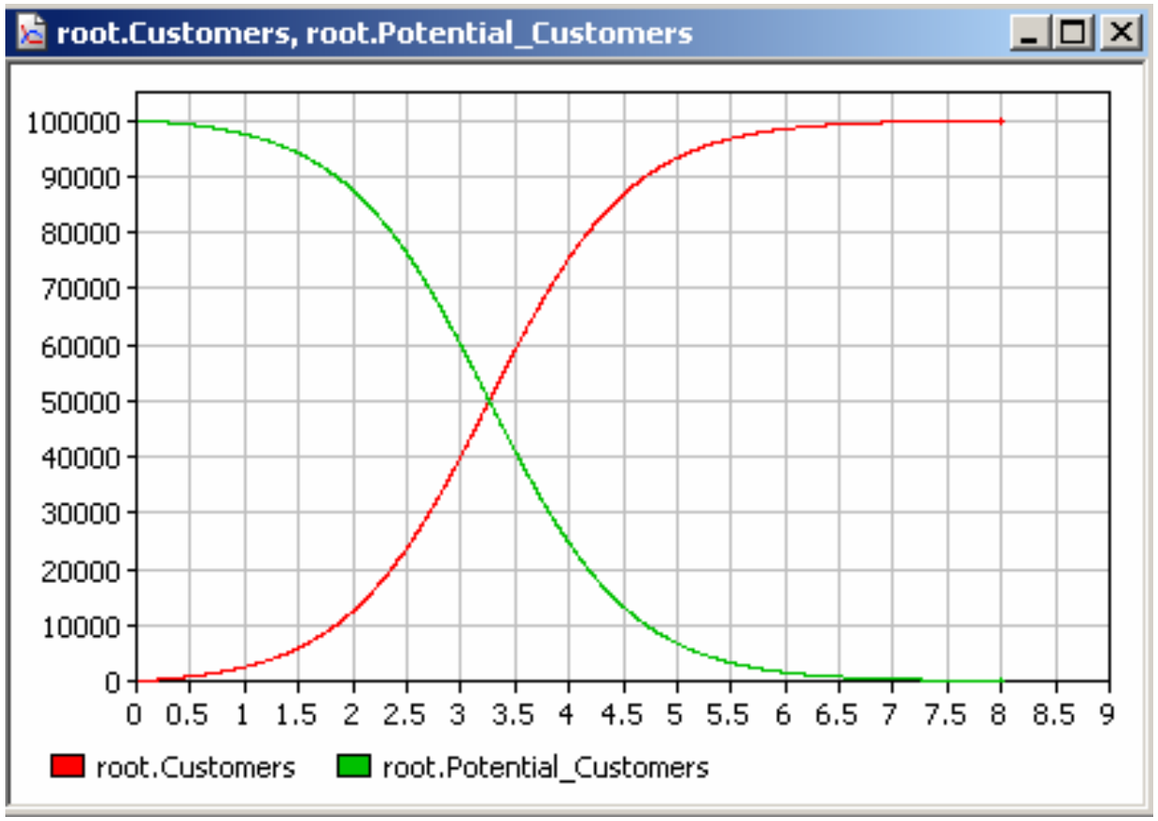


Окно оптимизации OptQuest

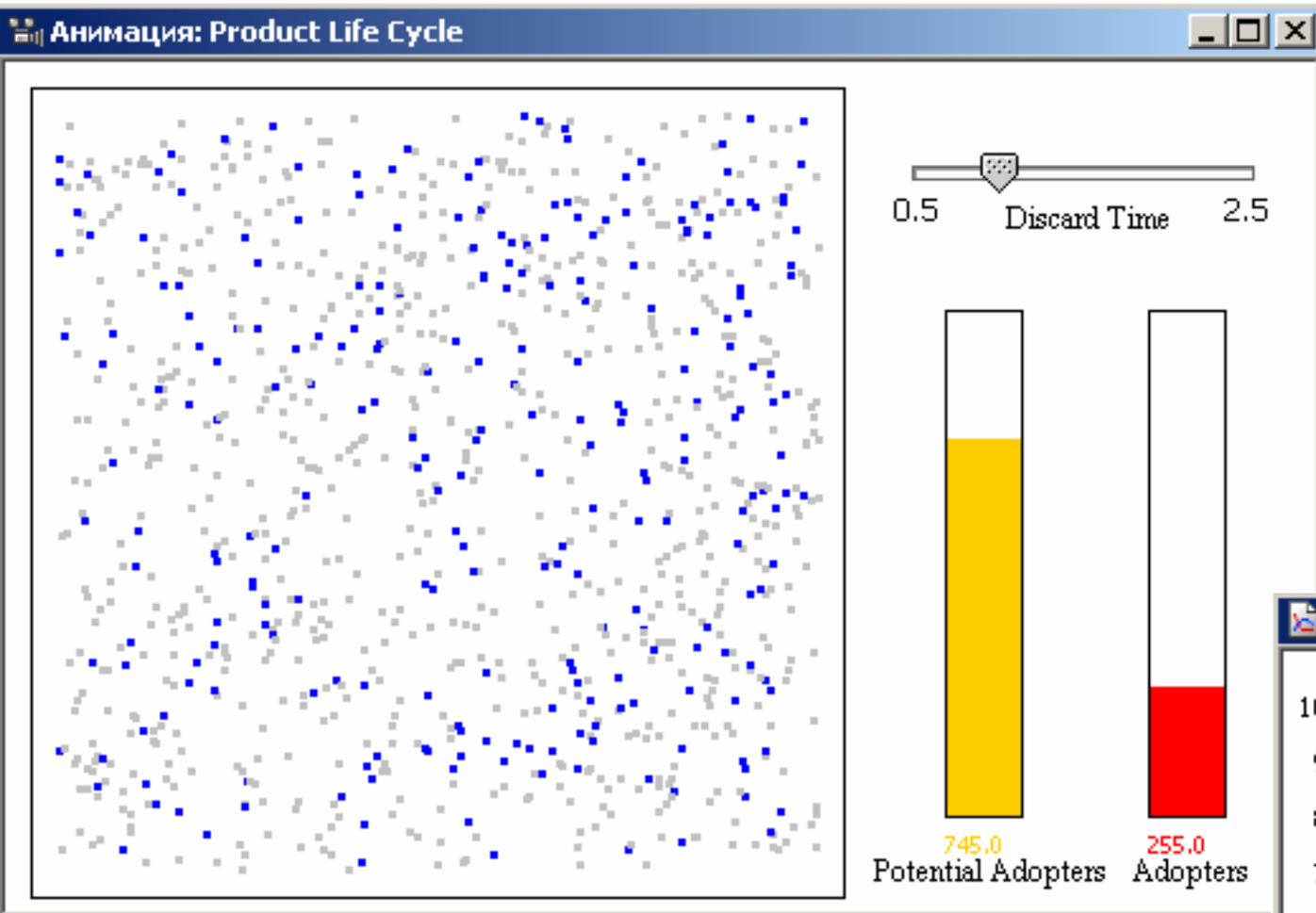


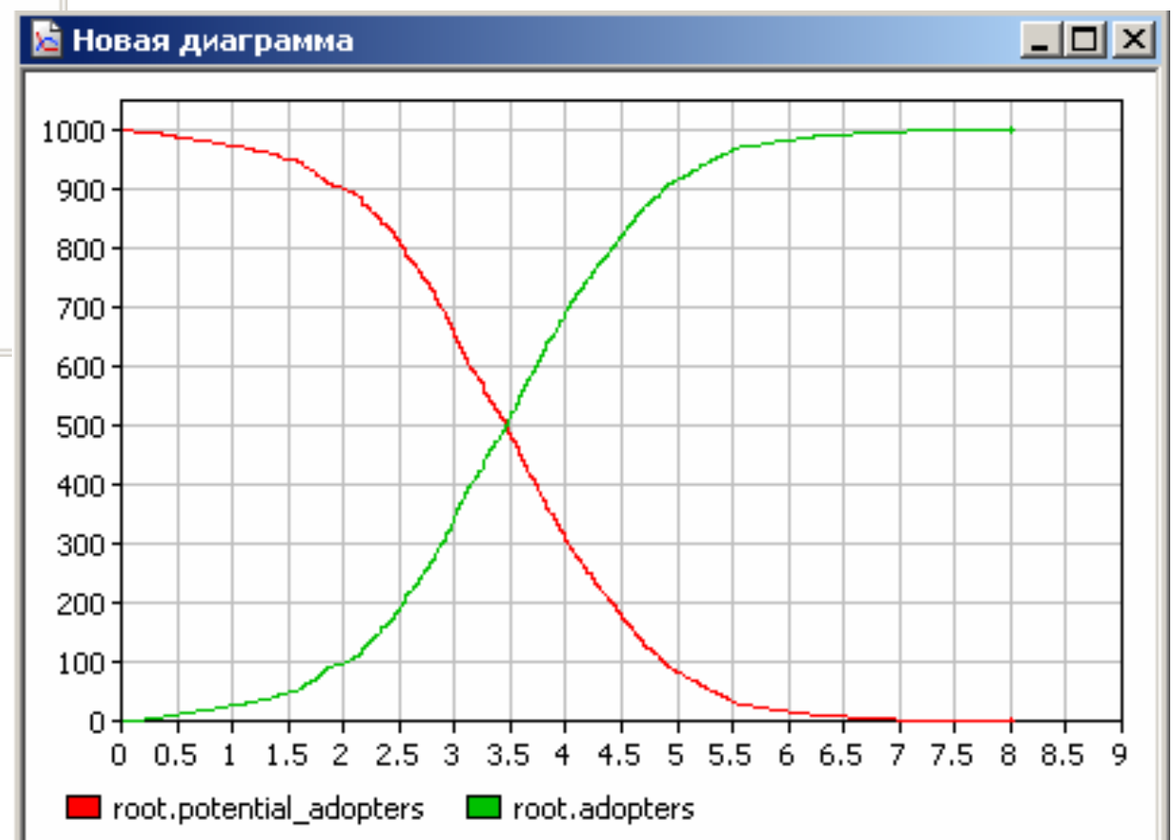
Системная динамика





Агентная модель





Справочные материалы по AnyLogic

- www.xjtek.ru
- Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб: БХВ-Петербург. 2005.390 с.
- Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика //Exponenta Pro, №3-4б 2004

Новые информационные технологии

В 80-х годах XX века предложен подход, в России получивший название новой информационной технологии [1, 2], на Западе — концепции ЭВМ V поколения [3].

1. **Поспелов Г.С. Искусственный интеллект - основа новой информационной технологии . М.: Наука, 1988. 279 с.**
2. **Глушков В.М. Основы безбумажной информатики. М.: Наука, 1982. 168 с.**
3. **Симонс Дж. ЭВМ пятого поколения, компьютеры 90-х годов. М.: Финансы и статистика,1985. 173 с.**

Суть новой информационной технологии (НИТ) — создание программных систем, работающих с конечным пользователем напрямую, минуя посредников в лице *математиков* и *программистов*.

Соглашение о допустимом минимуме профессиональных знаний пользователя

- ЛПР имеет представление об элементарной математике, хотя бы на уровне средней школы.
- ЛПР знает множество переменных рассматриваемой предметной области, а также имеет представление о границах значений, в которых эти переменные могут изменяться.
- ЛПР имеет представления о причинно-следственных или временных связях переменных.
- ЛПР знаком с компьютером хотя бы на уровне пользователя MS Office

Алгоритмическая модель - математическая модель исследуемого объекта, представленная в виде алгоритма, структуризация вычислительных процедур которого осуществлена в соответствии с представлениями пользователя о причинно-следственных или временных связях явлений, характеризующих моделируемый объект.

Язык алгоритмических сетей - графический язык, оперирующий исключительно функциональными зависимостями, имеющими место в предметной области, которой принадлежит исследуемый объект.

Основными понятиями языка являются понятия функционального оператора и интерфейсной дуги.

Алгоритмические сети

Алгоритмическая сеть (Рис.1(а)) определяется как ориентированный гиперграф без петель при вершинах G(V,X), в котором дуги обозначают модельные переменные $x_i \in X$,

а вершины $v_i \in V$ – функциональные соотношения (операторы), $f_i \in F$, связывающие модельные значения переменных на интервале времени, соответствующем Δt [2].

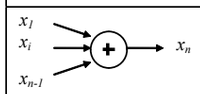
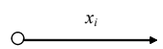
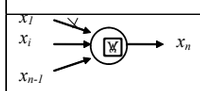
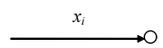
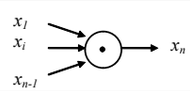
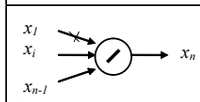
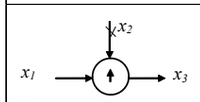
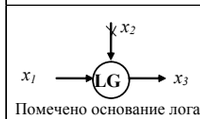
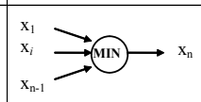
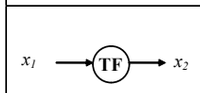
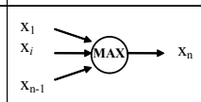
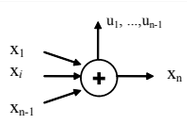
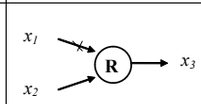
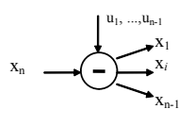
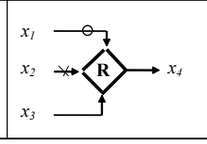
Структура переменных сети $X = X_{вх} \cup X_{вых} \cup X_{вн}$, где $X_{вх}$, $X_{вых}$, $X_{вн}$ – соответственно входные, выходные и внутренние переменные, причем, $X_{вх} \cap X_{вых} = \varnothing$, $X_{вх} \cap X_{вн} = \varnothing$, $X_{вых} \cap X_{вн} = \varnothing$.

Формально алгоритмическая сеть (АС) может быть определена следующим образом:

$$AC := \langle P, Q, X, F, P \to F, Q \to X \rangle$$

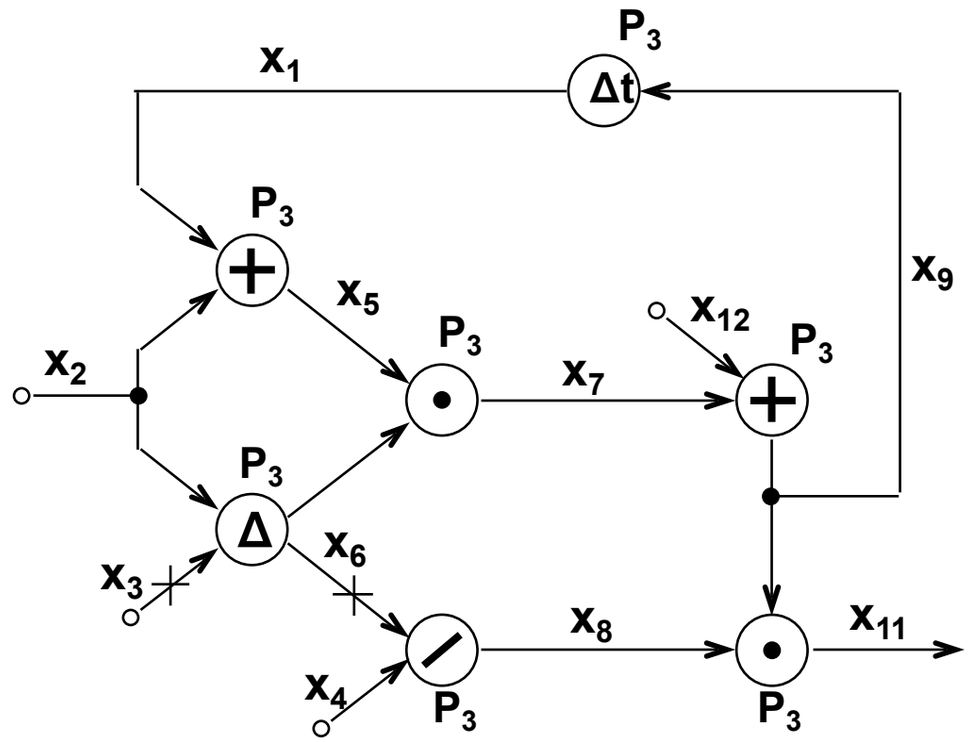
где: P – множество вершин сети, Q – множество дуг сети, X – множество переменных, при этом $X = \cup X_i$, X_i – множество переменных i-ой вершины, I – множество всех индексов вершин, F – множество операторов, $P \to F$ – изоморфное отображение P на F, $Q \to X$ – отображение Q на X.

Язык алгоритмических сетей

Операторы вычислители:			Операторы ввода-вывода:	
Графическое обозначение	**Реализуемое выражение**		**Графическое обозначение**	**Реализуемое выражение**
	x_n [illegible]			Ввод переменой
 Помечено уменьшаемое	x_n [illegible]			Вывод переменной
	x_n [illegible]		x_1 x_2	$x_2(t+1)=x_1(t)$ t - номер шага моделирования
 Помечено делимое	x_n [illegible]		**Логические операторы:**	
 Помечен показатель степени	$x_3 = x_1^{x2}$		**Графическое обозначение**	**Реализуемое выражение**
 Помечено основание логарифма	$x_3=\log_{x2} x_1$			$x_n=min(x_1,\ldots,x_{n-1})$
	$x_2=f_T(x_1)$ функция, задаваемая с помощью таблицы			$x_n= max(x_1,\ldots,x_{n-1})$
	$x_n=$ [illegible] x_i $u_i=x_i/x_1;\ i=1,n-1$			[illegible] x_1 R x_2 $x_3=$ [illegible] [illegible] x_1 [illegible] x_2 Помечен вход, стоящий слева от знака отношения R [illegible] =, ≠, <, ≤, >, [illegible]
	$u_{n-1}=1-$ [illegible] u_i $x_i=x_n$ [illegible] $u_1;\ i=1,n-1$			[illegible] x_1 R 0 $x_4=$ [illegible] [illegible] x_1 [illegible] 0 Помечен вход, значение которого присваивается выходу при истинности проверяемого условия R [illegible] =, ≠, <, ≤, >, [illegible]

Запрещённые конструкции:

- **Запрет вычисления одной и той же переменной более чем одним оператором («неоднозначность»).**
- **Запрет использования в качестве входной информации оператора, информации, являющейся результатом его расчета, если в образовавшемся контуре отсутствует оператор «задержка по времени («нецикл»).**
- **Запрет соединения альтернативных процессов моделирования в любых операторах сети, кроме оператора логического слияния («зависание»).**



P_1 P_5 P_2 P_6 P_3 P_7 P_4

P_1: $x_5(t)=x_1+x_2$
P_5: $x_6(t)=x_3-x_2$
P_2: $x_7=x_5 \cdot x_6$
P_6: $x_8=x_6/x_4$
P_3: $x_9=x_7+x_{12}$
P_7: $x_{11}=x_8 \cdot x_7$
P_4: $x_1(t+1)=x_9(t)$

а)

P_1 P_2

P_3 P_7 P_4
P_5 P_6

б)

Бинарные операции над АС

1. **Пересечение АС1∩АС2** равно подграфу, состоящему из вершин, совпадающих в обеих сетях .
2. **Объединение АС1∪АС2** равно пересечению АС и всем остальным вершинам той и другой АС . Особенность операции объединения АС - ее результат пуст, если при объединении нарушается условие однозначности вычисления переменных или возникают контура не содержащих вершин с оператором «задержка».
3. **Разность АС1\АС2** равна первой АС без подграфа, соответствующего пересечению исходных АС.

Все рассмотренные выше бинарные операции над АС сводятся к аналогичным теоретико-множественным операциям над множествами Fi алгоритмических сетей, над которыми операции выполняются.

Агрегация A{pi}(AC) соответствует замене некоторого подграфа исходной АС одной вершиной содержащей оператор, в котором объединены выражения, соответствующие всем операторам агрегируемого подграфа. Внешние дуги такой вершины соответствуют внешним дугам агрегируемого подграфа. {pi} – множество вершин агрегируемого подграфа.

Дезагрегация DApi(AC) операция обратная агрегации, соответствует замене некоторой вершины АС подграфом, построенным на основании выбора выражений, определяющих оператор, приписанный вершине pi.

Частичное обращение AC'=ObR(AC), результат операции новая АС с множеством входных переменных R. Сеть АС' строится путем обращения некоторых из операторов исходной сети на основании R и связей между вершинами.

Пример. *Исходный оператор x1+x2=x3.*

Обращение оператора для входных наборов:

R=(x1,x3) – набор допустим, обращенный оператор x3-x1=x2;

R=(x2) набор недостаточен, обращение невозможно;

R=(x1,x2,x3) набор избыточен - оператор переопределен, обращение невозможно.

Выделение подграфа по заданным подмножествам входных и выходных дуг AC'=S0I(AC). , где 0, I – множество переменных исходной АС, соответствующие которым дуги будут внешними выходными (входными) дугами получаемого подграфа.

Операции объединения и пересечения алгоритмических сетей ассоциативны и дистрибутивны друг относительно друга:

(AC1∪AC2) ∪AC3=AC1∪ (AC2∪AC3);

AC1∩AC2) ∩AC3=AC1∩ (AC2∩AC3);

(AC1∪AC2) ∩ AC3=(AC1∩AC2) ∪ (AC1∩AC3);

(AC1∩AC2) ∪ AC3=(AC1∪AC2) ∩ (AC1∪AC3);

Операция пересечения ассоциативна и дистрибутивна относительно операции пересечения:

(AC1\AC2) \AC3=AC1\ (AC2\AC3);
(AC1\AC2) ∩ AC3=(AC1∩AC2) \ (AC1∩AC3);

(AC1∩AC2) \ AC3=(AC1\AC2) ∩ (AC1\AC3);

Операции пересечения и объединения коммутативны и идемпотентны:

AC1∪AC2=AC2∪AC1, AC1∩AC2=AC2∩AC1,

AC1∪AC1=AC1, AC1∩AC1=AC1.

Правила поглощения:

(AC1∩AC2) ∪AC1=AC1; (AC1∪AC2) ∩AC1=AC1;

(AC1\AC2) ∪AC1=AC1; (AC1∪AC2) \AC1=AC2;

(AC1∩AC2) \AC1=∅; AC1/AC1=∅;

(AC1\AC2) ∩AC1=AC1\AC2; (AC1\AC2) ∪AC2=AC1∪AC2.

Операции с пустой сетью ∅ :

AC1∪∅=∅∪AC1=AC1; AC1∩∅=∅∩AC1=AC1;

Если считать, что все АС есть подсети всеобщей сети ***АС,*** то такая сеть будет единицей для пересечения и нулем для объединения:

$$AC1 \cup \boldsymbol{AC} = \boldsymbol{AC} \cup AC1 = AC; \quad AC1 \cap \boldsymbol{AC} = \boldsymbol{AC} \cap AC1 = AC1;$$

Дополнение некоторой АС до всеобщей обозначим ⌉ АС :

$$\neg AC = \boldsymbol{AC} \setminus AC.$$

Свойства дополнения и правило Де Моргана для АС:

$$\neg(\neg AC) = AC,$$

$$\neg(AC1 \cup AC2) = \neg AC1 \cap \neg AC2, \quad \neg(AC1 \cap AC2) = \neg AC1 \cup \neg AC2;$$

$$\neg \boldsymbol{AC} = \varnothing,$$

$$AC \cup \neg AC = \boldsymbol{AC}, \quad AC \cap \neg AC = \varnothing, \quad AC \setminus \neg AC = AC, \quad \neg AC \setminus AC = \neg AC,$$

$$AC \cup \neg AC = \boldsymbol{AC}.$$

Множество всех алгоритмических сетей и заданные операции определяют алгебру АС :

$$A = \langle \{AC\}, \Omega \rangle$$

Где {АС} – множество всех возможных АС, Ω - множество операций над АС.

Согласование фрагментарных моделей

Исходное множество фрагментарных одноуровневых моделей предметной области:

- общий словарь переменных,
- синтаксическая правильность моделей отдельных фрагментов.

Результат формального объединения моделей:

- синтаксически правильная комплексная модель,
- синтаксически ошибочная комплексная модель в результате:
  - неоднозначного вычисления переменных,
  - образования некорректных циклов,
  - потери вычислимости.

Причины рассогласованности моделей исходного множества:

- использование различных алгоритмов вычисления одноименных переменных,
- каузальная рассогласованность вычисления переменных.

Выявление каузальной рассогласованности

Выявление каузальной рассогласованности:
-построение сокращенного графа причинно-следственных связей на множестве фрагментарных моделей,
- переход к древовидному графу,
-сравнение цепочек вычисления переменных модели с цепочками событий на графе

$$CT_j, \bigcup CT_j = CT \quad j = \overline{1,s},$$

Ro(x,y) - событие x происходит одновременно с y,
R1(x,y) - событие x происходит раньше y,
R2(x,y) - событие x происходит не раньше y.
y_k - общая вычисляемая переменная сетей A и B
X1 и Z1 - множества переменных сети A, входящих в конус и дерево вычислимости для y_k,
X2 и Z2 - множества переменных сети B, входящих в конус и дерево вычислимости для y_k.
Тогда: R1(X1,y_k), R1(y_k,Z1), R1(X1,Z1), R1(X2,y_k), R1(y_k,Z2), R1(X2,Z2).

Проверка данных соотношений на цепочках событий CT_j.
Если выполняется условие:

$$(\forall x \in X)(\forall z \in Z)(\forall CT_j \in CT)[(x,z \in CT_j \rightarrow R1(x,z))\&$$

$$\&(x,yk \in CTj \rightarrow R1(x,yk))\ \&(yk,z \in CTj \rightarrow R1(yk,z))],$$

то сеть каузально ориентирована, в противном случае – инверсно ориентирована.

Корректировка моделей

Операции корректировки фрагментарных моделей:

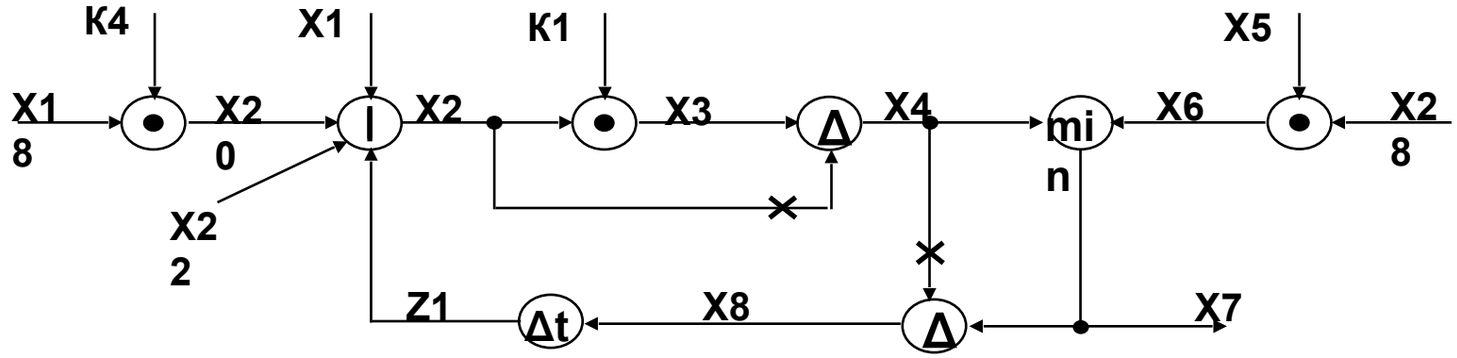
- каузальное согласование фрагментов с использованием операции обращения для устранения неоднозначности вычисления переменных и восстановления вычислимости,
- операции вычитания для устранения неоднозначности вычисления переменных в каузально согласованных фрагментах,
- корректировка циклов для устранения «гонок» в комплексной модели.

Операция вычитания AΔB=C. Из сети А удаляются операторы, генерирующие одноименные вычисляемые переменные сетей А и В, **а также операторы конусов вычислимости этих переменных, за исключением операторов, генерирующих переменные, которые являются входными для сети В, являются аргументами оставшихся в сети А операторов или имеют статус критериальных. Вместе с операторами удаляются их выходные переменные. В результате обеспечивается взаимная однозначность В и результирующей сети С.**

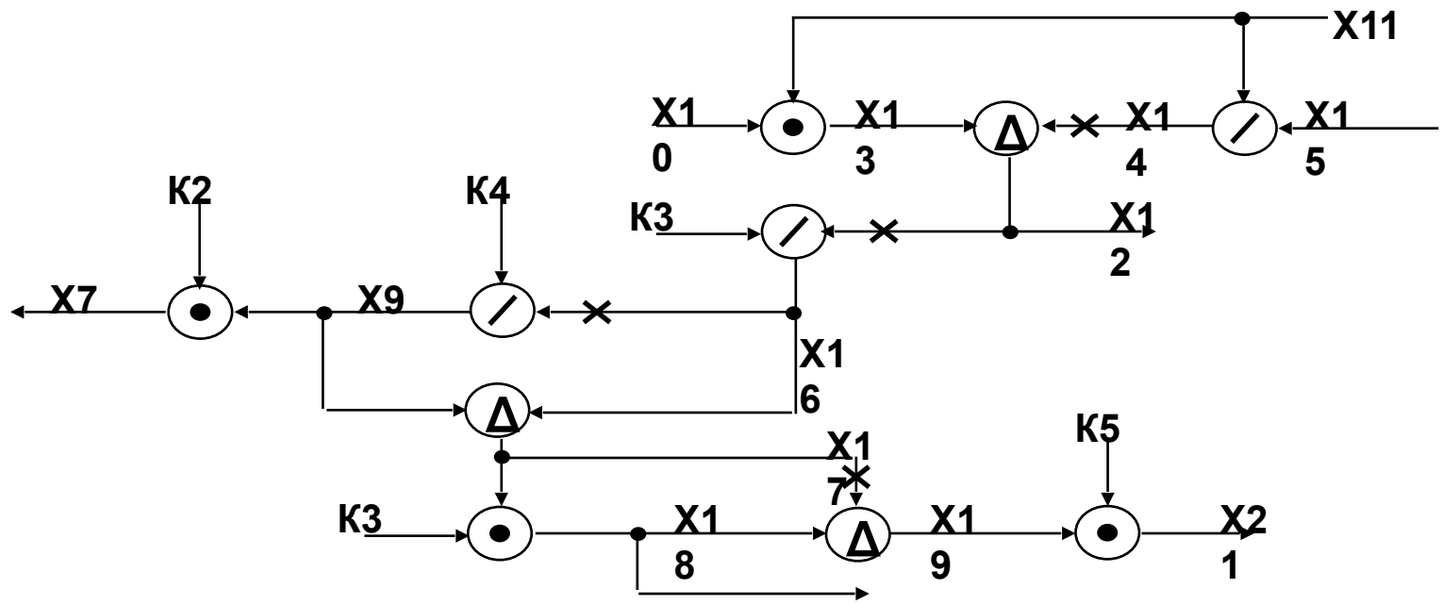
Корректировки не-циклов. Особенность процедуры состоит в ее неоднозначности, поскольку статус переменной состояния может быть присвоен различным переменным контура.

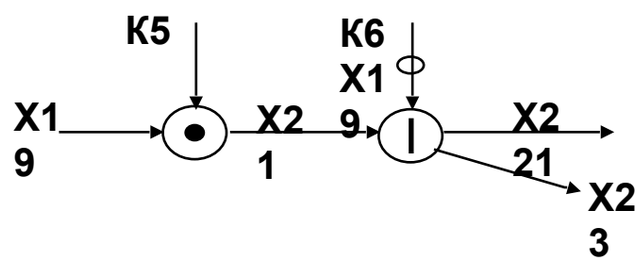
Численность закупаемых животных
X11
Общий прирост биомассы
X12
Конечный вес животных
X15
Животноводческая ферма
Очистные сооружения
Начальный вес животных
X10
X19
Жидкие стоки
Кормовые корнеплоды
X7
Органические удобрения
X18
X22
X23
Поле
Минеральные удобрения
X1
Жесткая фракция стоков
Стоки с полей
X3
Озеро
Вынос фосфора с речным стоком
X26
Очищенные стоки
X27
Захоронения фосфора в донных отложениях

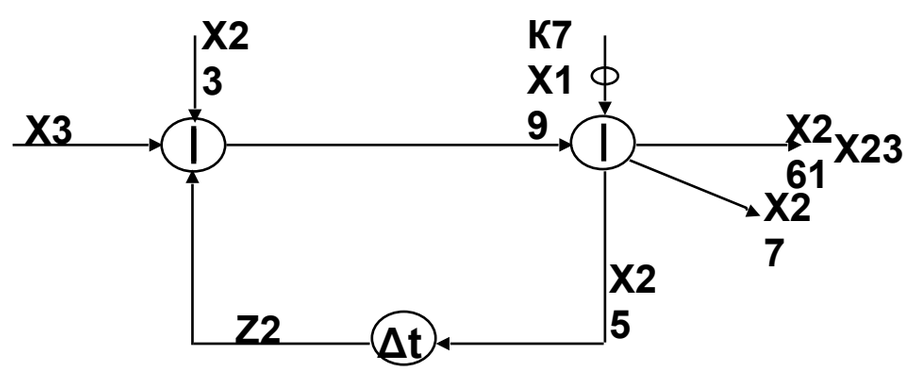
1.
2.
3.

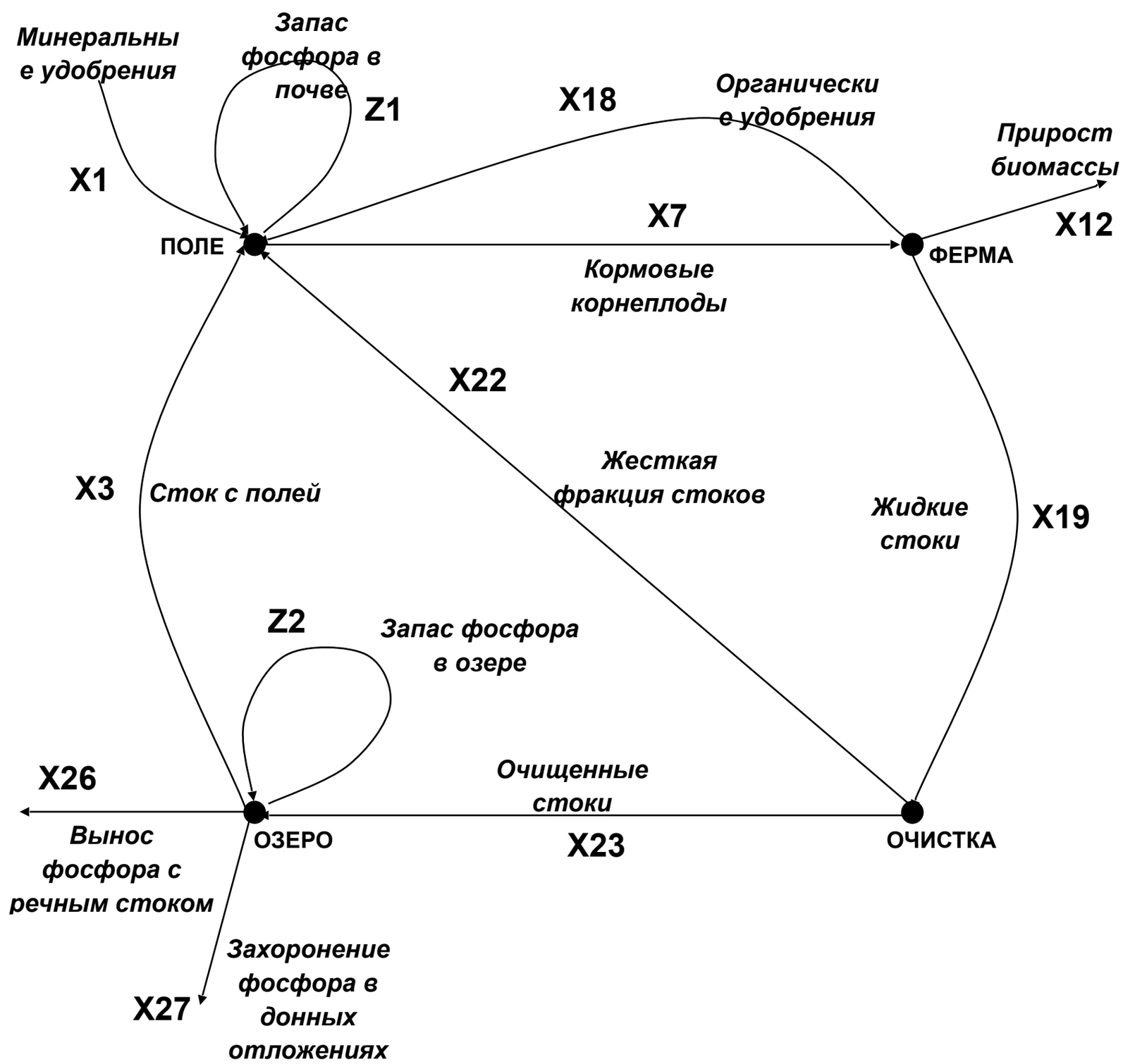

K4
X1
K1
X5
X18
X20
X2
X3
X4
min
X6
X28
X22
Z1
Δt
X8
X7

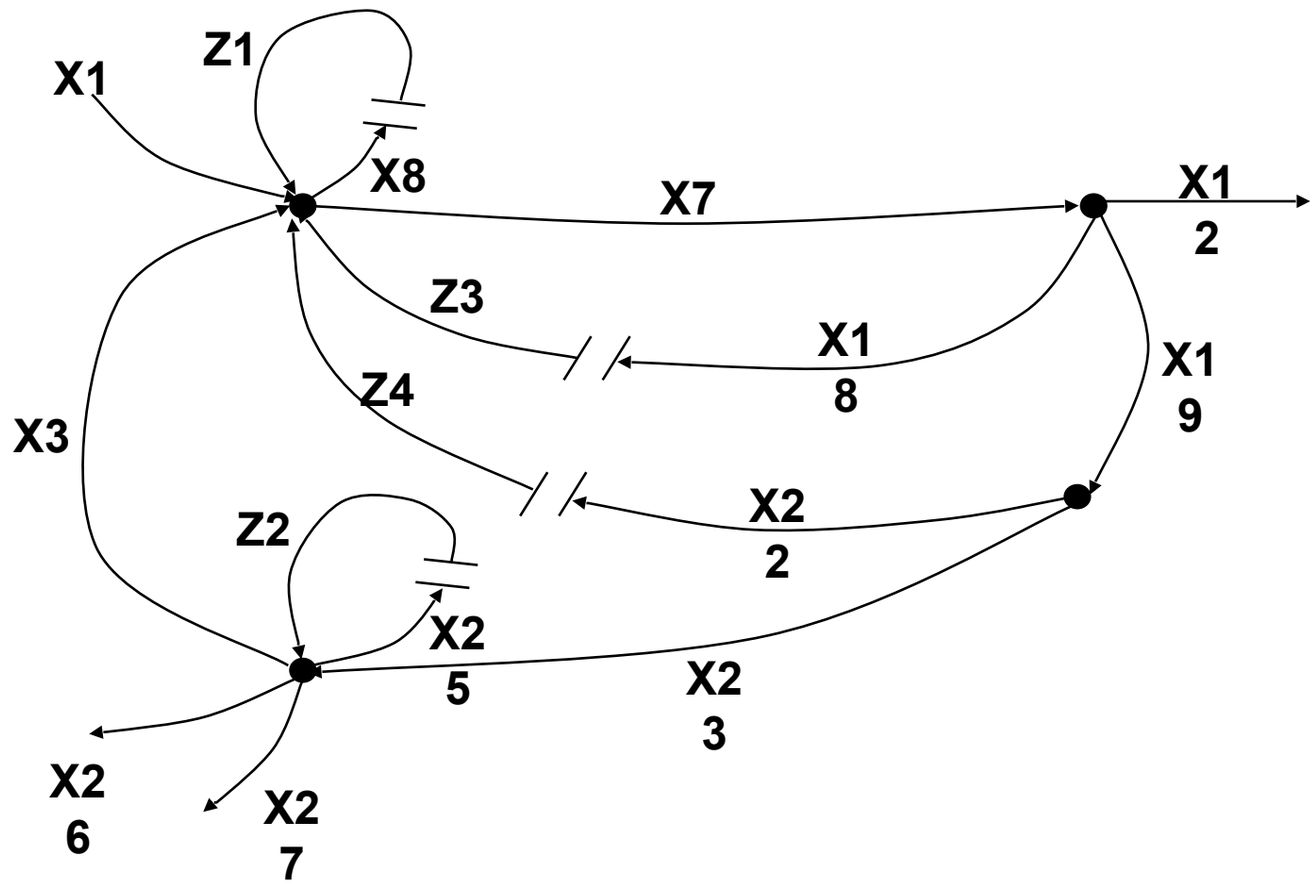
4.

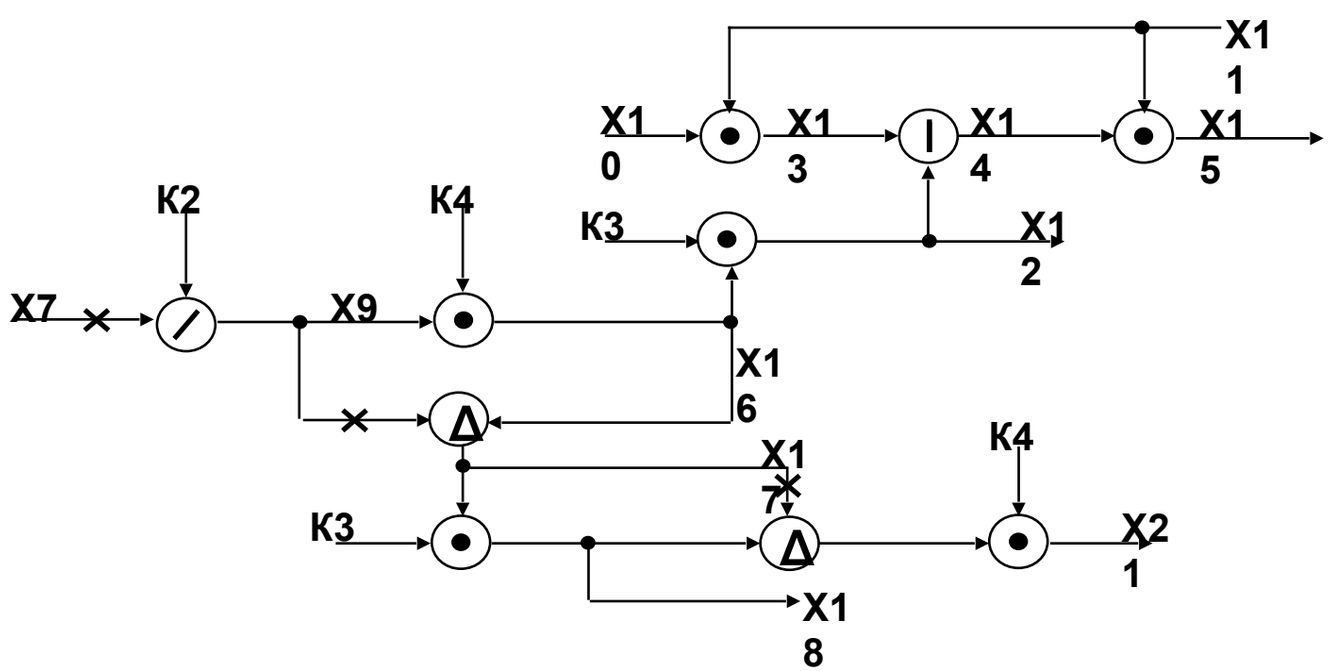

X11
X10
X13
X14
X15
K2
K4
K3
X12
X7
X9
X16
X17
K5
K3
X18
X19
X21

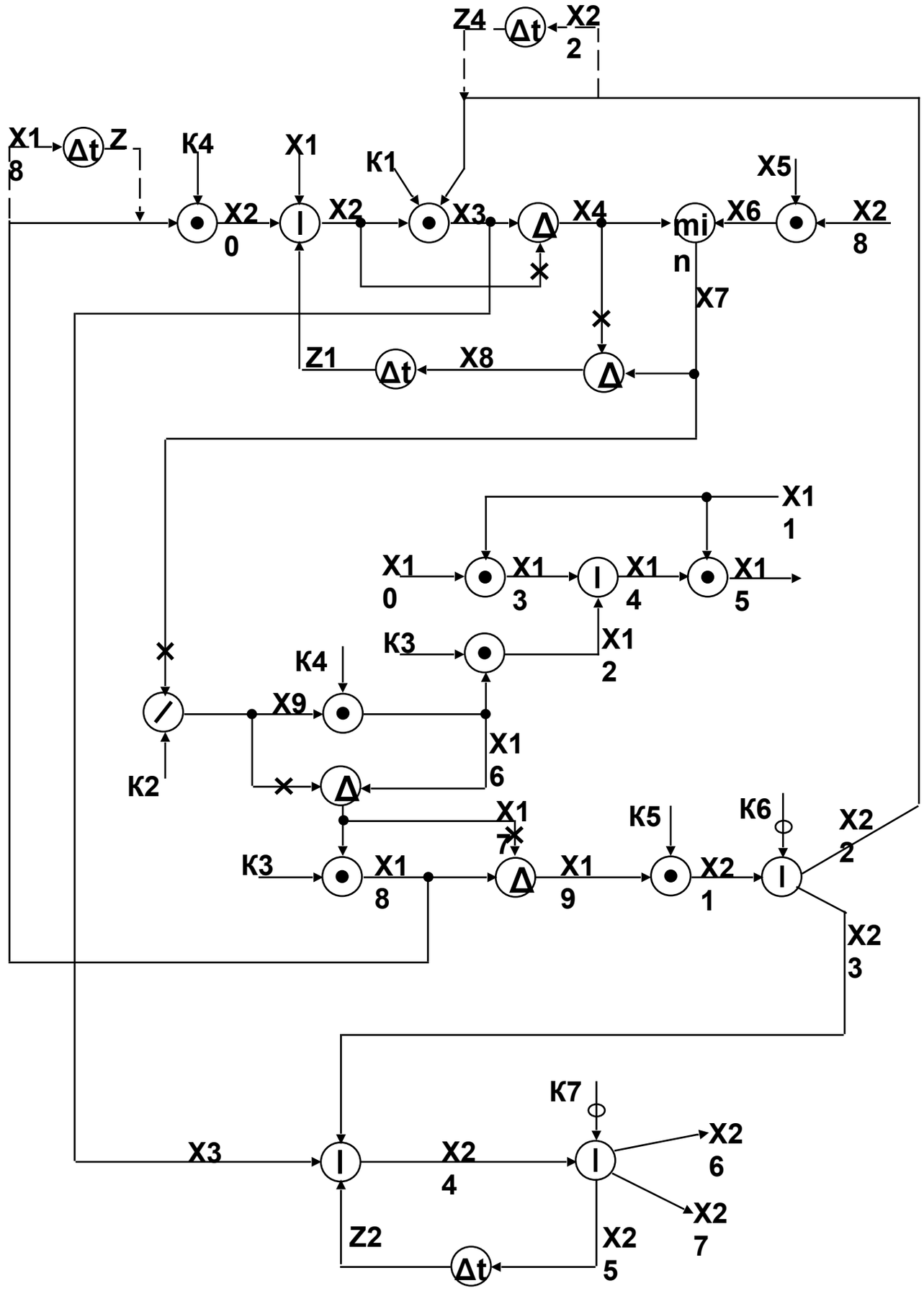

K5
K6
X19
X19
X21
X21
X23


X23
K7
X19
X3
X261
X23
X27
X25
Z2
Δt


Минеральны
е удобрения
Запас
фосфора в
почве
Z1
X18
Органически
е удобрения
Прирост
биомассы
X1
X7
X12
ПОЛЕ
ФЕРМА
Кормовые
корнеплоды
X22
Жесткая
фракция стоков
X3
Сток с полей
Жидкие
стоки
X19
Z2
Запас фосфора
в озере
X26
Очищенные
стоки
Вынос
фосфора с
речным стоком
ОЗЕРО
X23
ОЧИСТКА
Захоронение
фосфора в
донных
отложениях
X27


X1
Z1
X8
X7
X1
2
Z3
X1
8
X1
9
Z4
X3
X2
2
Z2
X2
5
X2
3
X2
6
X2
7


X1
1
X1
0
X1
3
X1
4
X1
5
K2
K4
K3
X1
2
X7
X9
X1
6
X1
7
K4
K3
X2
1
X1
8


Z4
Δt
X2
2
X1
8
Δt
Z
K4
X1
K1
X5
X2
0
X2
X3
X4
mi
n
X6
X2
8
X7
Z1
Δt
X8
X1
1
X1
0
X1
3
X1
4
X1
5
K3
X1
2
K4
X9
X1
6
K2
X1
7
K5
K6
X2
2
K3
X1
8
X1
9
X2
1
X2
3
K7
X3
X2
4
X2
6
X2
7
Z2
X2
5
Δt

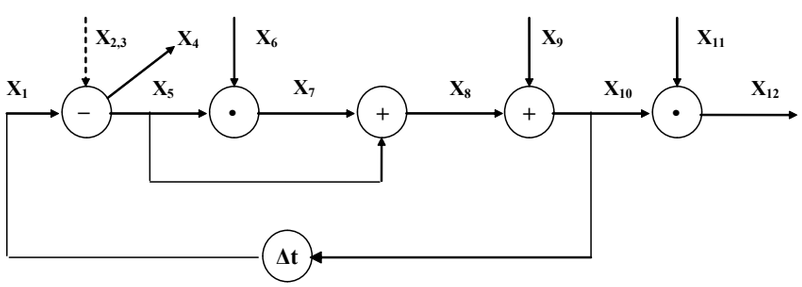
Методология моделирования на основе АС

Концептуальная модель:

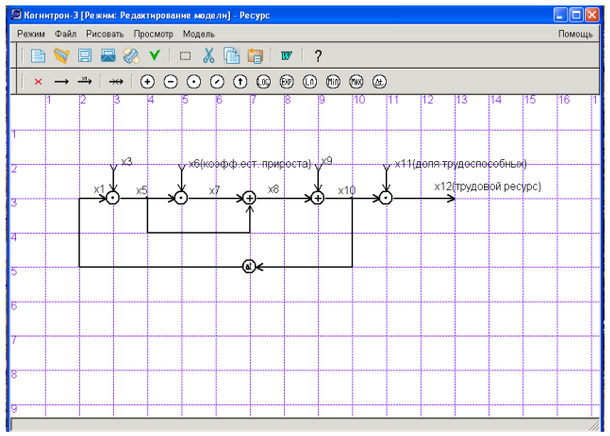
Язык алгоритмических сетей:

Î ï åðàòî ðû âû ÷è ñë è òåë è :		Î ï åðàòî ðû ââî äà-âû âî äà:	
Ãðàô è ÷åñêî å î áî çí à÷áí è å	Ðåàë è çóåì î å âû ðàæåí è å	Ãðàô è ÷åñêî å î áî çí à÷áí è å	Ðåàë è çóåì î å âû ðàæåí è å
x_1, x_i, x_{n-1} → + → x_n	$x_n = \sum_{1}^{n-1} x_i$	x_i	Ââî ä ï åðåì åí î é
x_1, x_i, x_{n-1} → Δ → x_n; Ï î ì å÷áí î óì åí üø ååì î å	$x_n = x_1 - \sum_{2}^{n-1} x_i$	x_i	Âû âî ä ï åðåì åí í î é
x_1, x_i, x_{n-1} → • → x_n	$x_n = \prod_{1}^{n-1} x_n$	x_1 → Δt → x_2	$x_2(t+1)=x_1(t)$ t - í î ì å ðð ø àãà ì î äåëèðî âàí èÿ
x_1, x_i, x_{n-1} → / → x_n; Ï î ì å÷áí î äåëèì î å	$x_n = x_1 / \prod_{2}^{n-1} x_n$	Ëî ãè÷åñêè å î ï åðàòî ðû :	

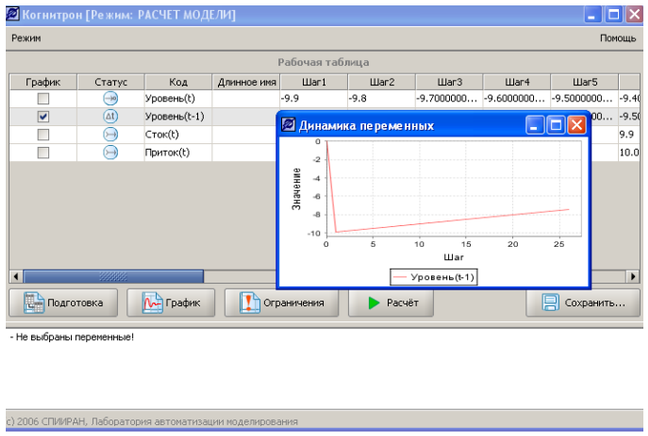
Алгоритмическая сеть модели:



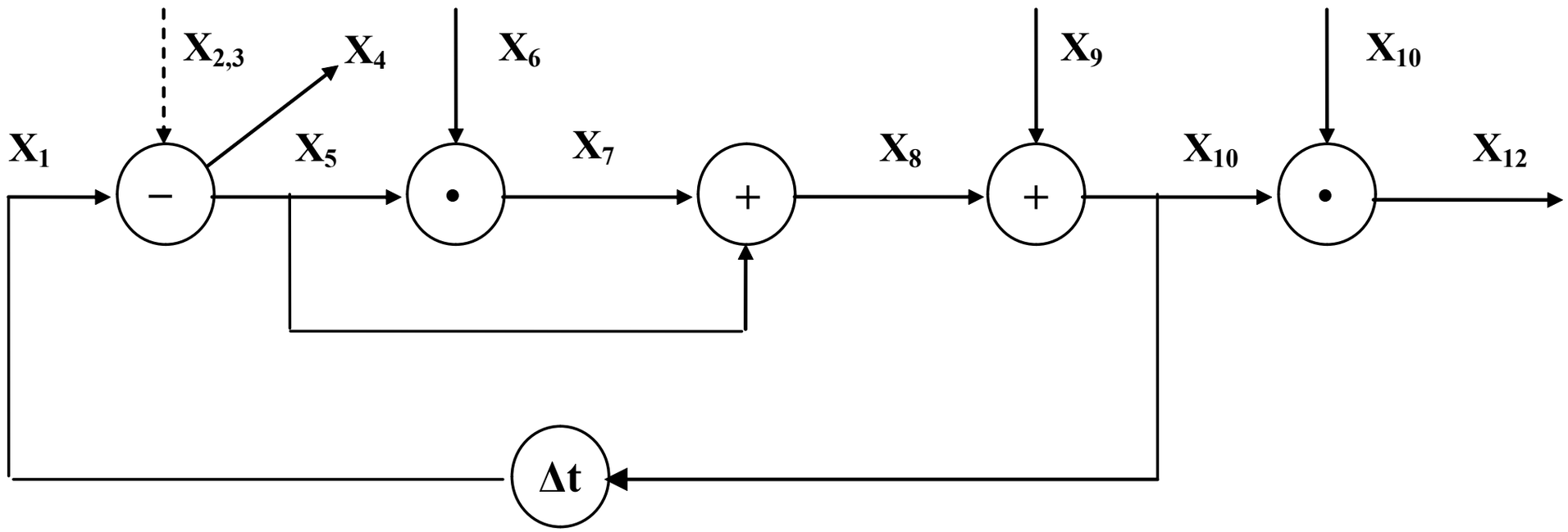
Создание модели:

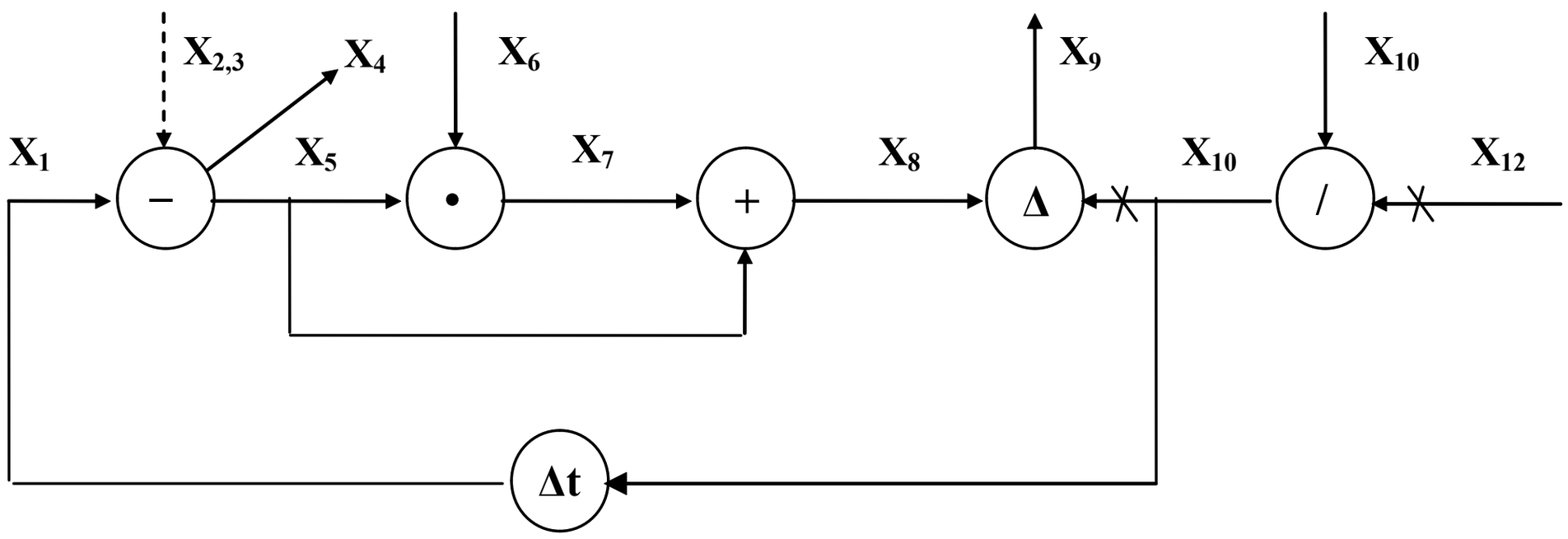


Модельные эксперименты:

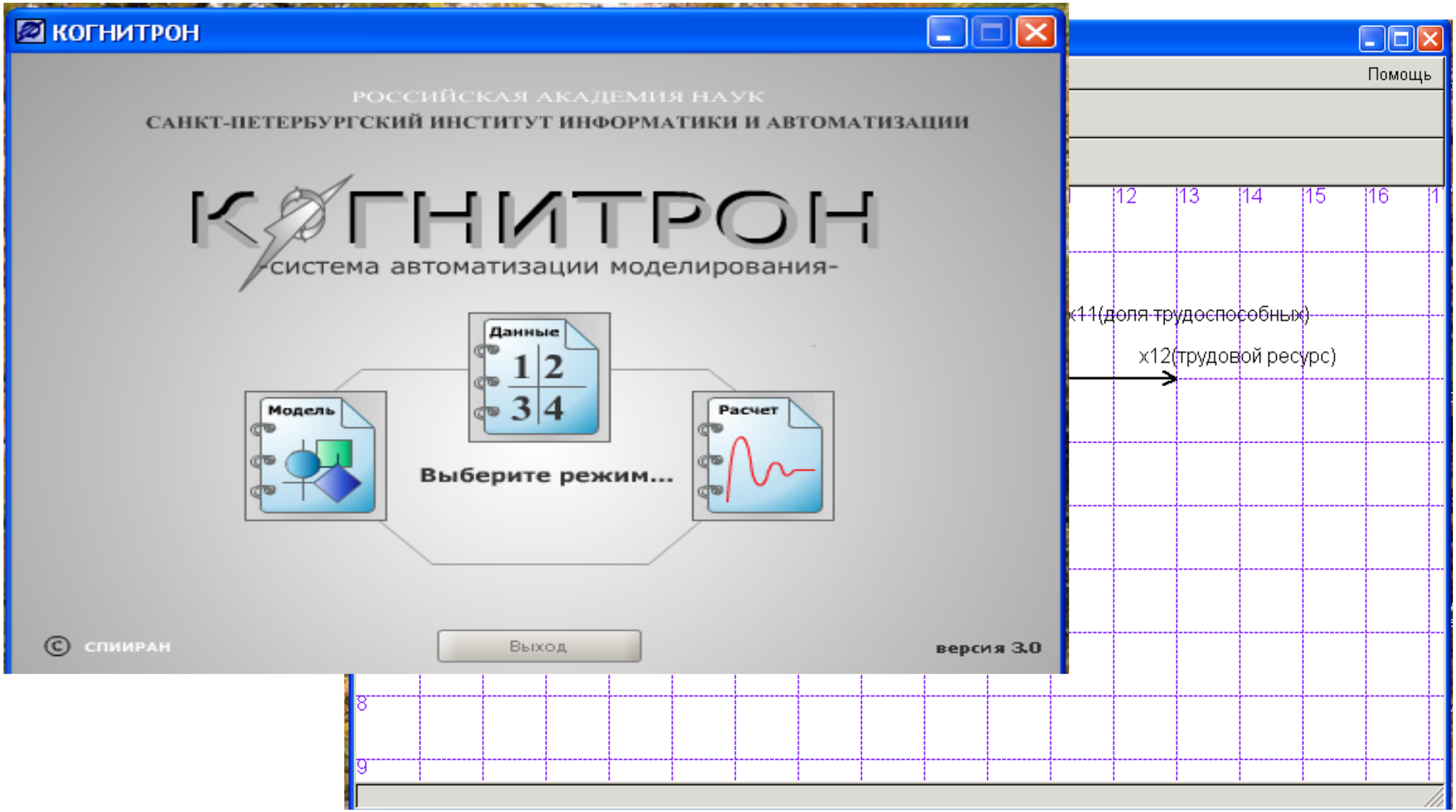


Пример решения задачи методом обращения

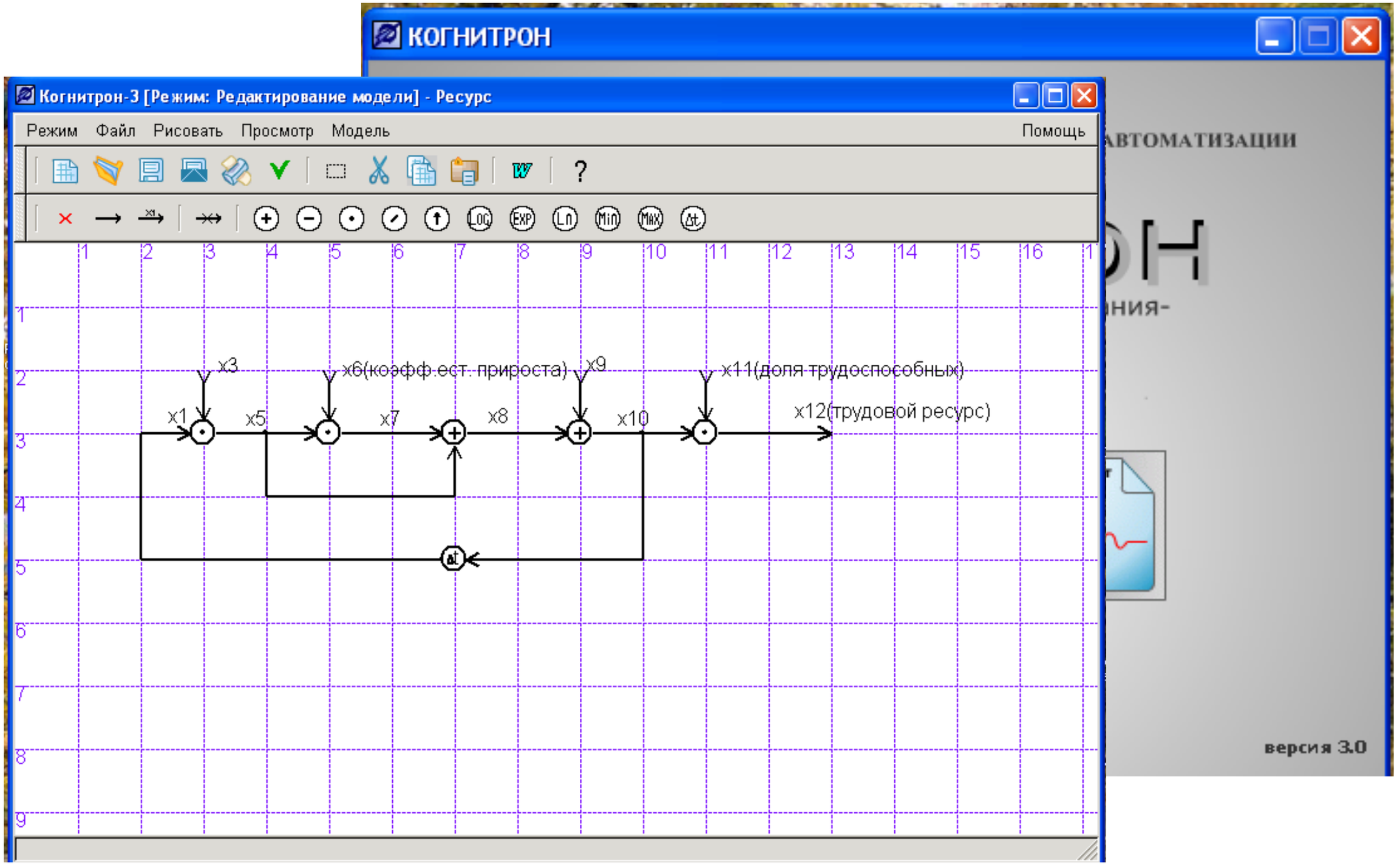




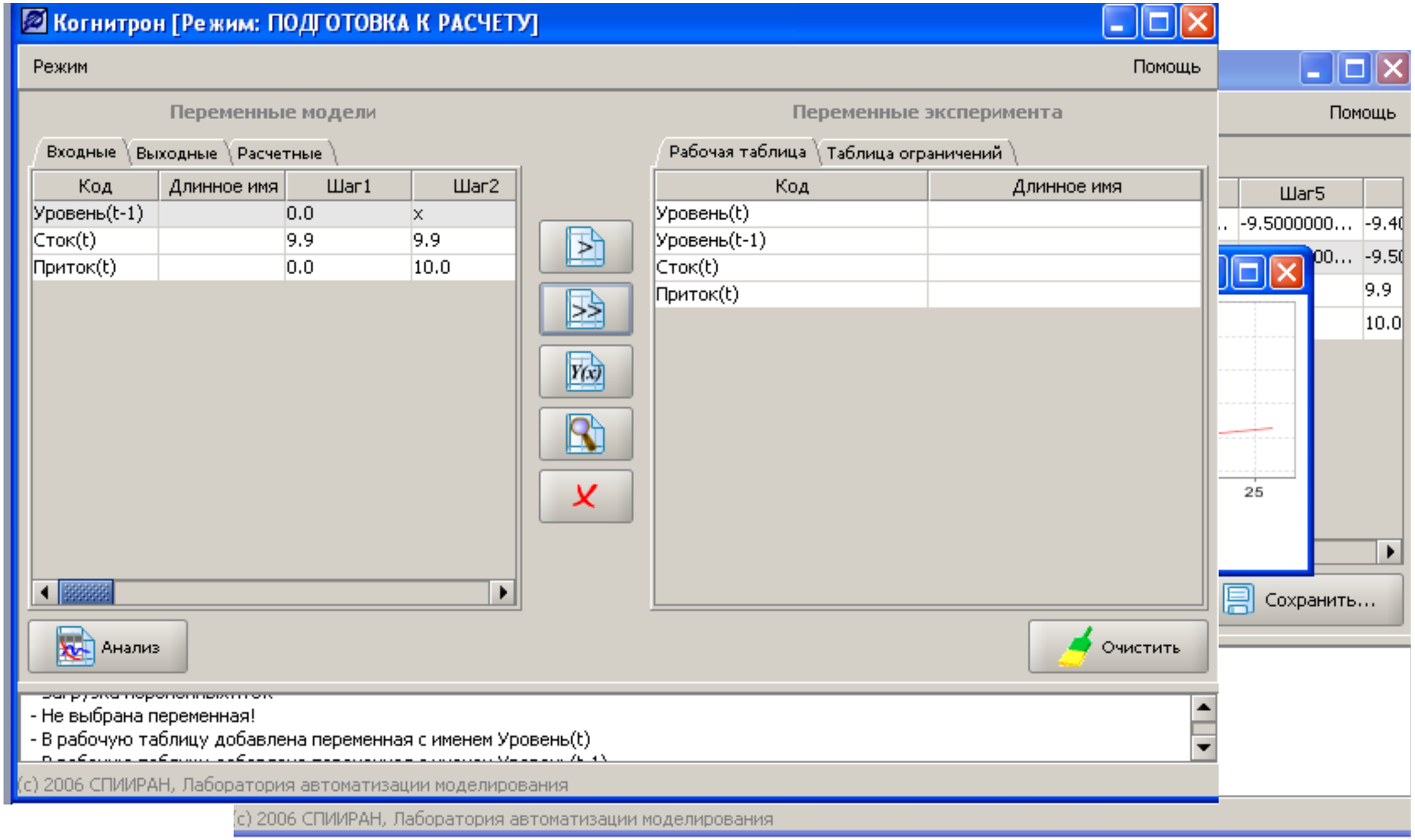
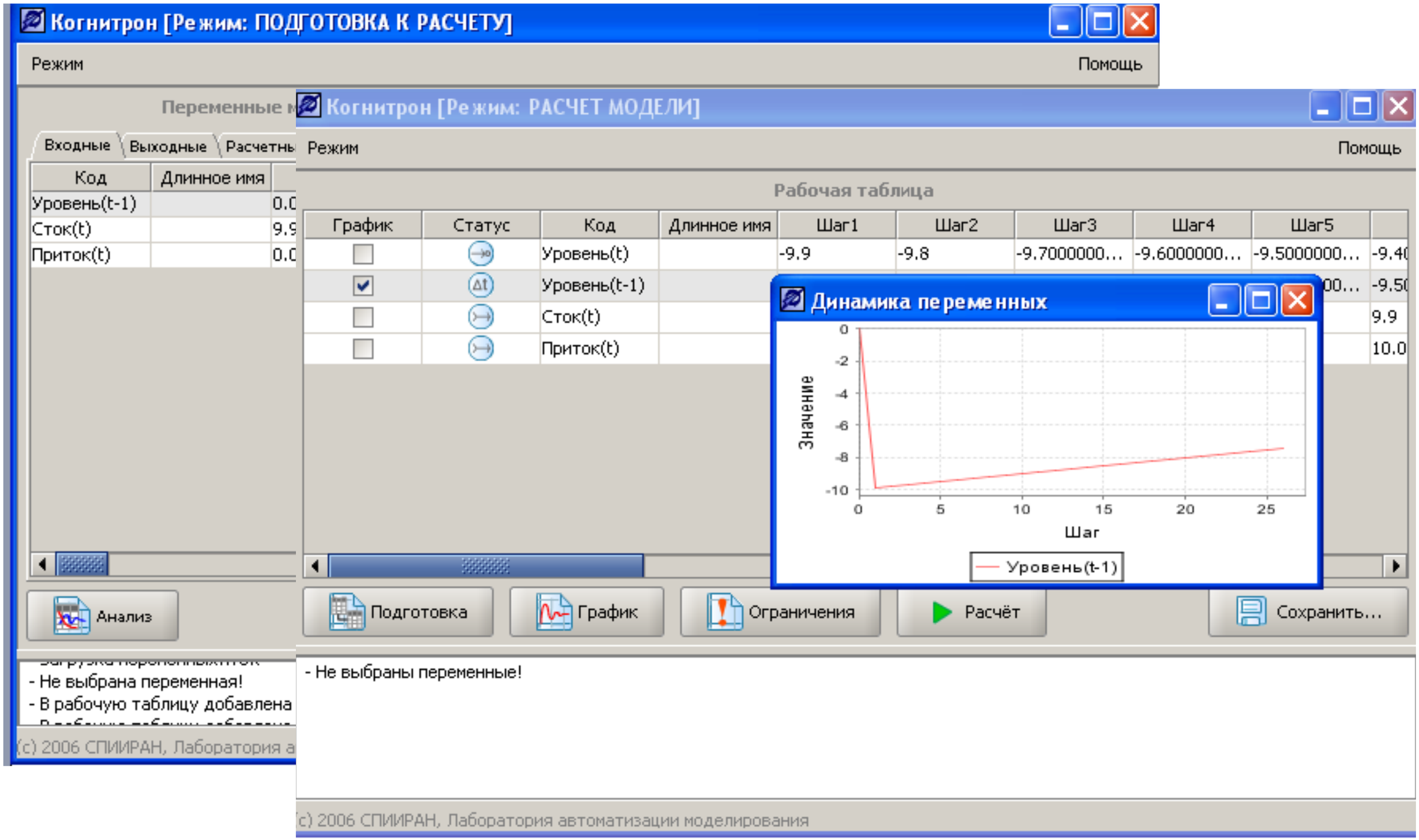
Система КОГНИТРОН.
Создание модели



Система КОГНИТРОН.
Создание модели



Система КОГНИТРОН.
Модельный эксперимент



Система КОГНИТРОН.
Модельный эксперимент