

Е.В. Худякова, А.А. Липатов

**ИМИТАЦИОННОЕ
МОДЕЛИРОВАНИЕ ПРОЦЕССОВ
И СИСТЕМ В АПК**

Учебник

Москва 2021

УДК 338.24.01
ББК 65.050
Х 982

Рецензенты:

С.О. Сиптиц

Зав. отделом системных исследований экономических проблем АПК Всероссийского института аграрных проблем и информатики имени А.А. Никонова – Филиал ФГБНУ «ФИЦ ВНИИЭСХ», доктор экономических наук, доцент

В.И. Меденников

Ведущий научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской Академии наук (ФИЦ ИУ РАН), доктор технических наук

Худякова Е.В., Липатов А.А.

Х **Имитационное моделирование процессов и систем в АПК/**
98 Учебник. – М: ИКЦ «Колос-с», 2021 -256 с.: ил..

ISBN 004.94:338.3(075.8)

ББК 65.32вбя73

Х98

Учебник разработан в соответствии с требованиями государственного образовательного стандарта по дисциплине «Имитационное моделирование» для специальности «Прикладная информатика (в экономике)». В учебнике изложены общие вопросы имитационного моделирования, а также способы построения имитационных моделей с помощью программ Vensim, GPSS World и AnyLogic.

Содержание

Введение.....	6
ГЛАВА 1. СУЩНОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ.....	8
1.1 Понятие математического моделирования.....	8
1.2. Сущность метода имитационного моделирования.....	10
1.3. Представление времени в имитационной модели	14
1.4. Классификация имитационных моделей.	16
1.5. Экспериментальная природа имитационного моделирования.....	17
1.5.1. Метод статистических испытаний Монте-Карло	17
1.5.2. Понятие имитационного эксперимента	22
1.6. Этапы имитационного моделирования	24
1.7. Области применения имитационного моделирования.....	25
1.8. Программные средства создания имитационных моделей.....	27
Контрольные вопросы к главе I.....	29
ГЛАВА 2. ДИСКРЕТНО-СОБЫТИЙНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В АПК.....	31
2.1. Системы массового обслуживания и их сети.....	31
2.2. Основные элементы GPSS World	36
2.3. Типы операторов GPSS World	38
2.4. Имитационный проект.....	52
2.4.1. Формулирование и описание проблемы (постановка задачи)	52
2.4.2. Создание концептуальной модели	53
2.4.3. Формализация имитационной модели	56
2.4.4. Подготовка исходной информации	57
2.4.5. Программирование модели	58
2.4.6. Анализ и интерпретация результатов пробного прогона модели.....	60
2.4.7. Исследование свойств имитационной модели.....	66
2.4.7.1. Оценка адекватности имитационной модели и точности результатов моделирования	66

2.4.7.2. Определение объема выборки (числа прогонов)	68
2.4.7.3. Оценка устойчивости модели	69
2.4.8. Планирование и проведение имитационного эксперимента	73
2.4.8.1. Эксперимент ANOVA (дисперсионный анализ)	74
2.4.8.2. Оптимизационный эксперимент	79
Контрольные вопросы к главе II	84
Задания для разработки имитационного проекта в системе GPSS World	85
ГЛАВА III. МЕТОД СИСТЕМНОЙ ДИНАМИКИ	99
3.1. Основные понятия метода системной динамики	99
3.2. Событийно-ориентированный подход к моделированию систем	100
3.3. Понятие и виды обратных связей в социально-экономических системах	102
3.4. Особенности моделирования сложных динамических систем	107
3.5. Этапы имитационного моделирования методом системной динамики	110
3.6. Основные элементы модели системной динамики	111
3.7. Основное меню системы Vensim	118
3.8. Панель управления файлом модели, прогонами и настройками модели	125
3.8.1. Управление файлами и панель настройки	125
3.8.2. Графики и таблицы пользователя	138
3.9. Панель инструментов для построения модели	141
3.10. Импорт данных в модель	142
3.11. Пример построения модели в системе Vensim PLE	144
3.12. Функция LOOKUP	154
3.14. Оценка чувствительности моделей системной динамики	161
3.15. Оптимизационный эксперимент в моделях системной динамики	167
3.14.1. Модель «Зависимость динамики прибыли предприятия от схемы погашения долга»	168
3.14.2. Модель «Выбор оптимальной стратегии использования ресурсов»	172
3.14.3. Модель «Стратегии кормопроизводства сельскохозяйственного предприятия»	176
3.14.4. Модель «Динамика распространения продукции предприятия на рынке региона»	181

3.14.6. Модель «Оценка вариантов инвестиционного проекта строительства цеха по переработке молока сельскохозяйственным предприятием».....	183
3.14.6. Модель «Прогнозирование объемов валовой продукции сельского хозяйства региона и доходов населения»	189
Контрольные вопросы к главе III	195
ГЛАВА 4. АГЕНТНОЕ МОДЕЛИРОВАНИЕ	196
4.1. Сущность агентного моделирования процессов и систем.....	196
4.2.Имитационная модель уборки урожая в сельскохозяйственном предприятии.....	203
4.2.1. Постановка задачи.....	204
4.2.2. Порядок создания модели	204
4.2.2.1. Создание агентов.....	205
4.2.2.2. Настройка анимации агентов.....	206
4.2.2.3. Настройка логических процессов.....	209
4.2.2.4. Создание диаграмм процессов.....	217
4.2.2.5. Добавление в модель элементов сбора статистики	218
о характеристиках бизнес-процесса	218
3.2.2.6. Создание элементов управления и прогон модели.....	220
4.2.3. Проведение оптимизационных экспериментов на модели.....	224
Контрольные вопросы к главе IV.....	226
ГЛАВА 5. ПРИМЕНЕНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ.....	228
5.1. Имитационное моделирование и облачные технологии.....	228
5.2. GPSS Cloud и AnyLogic Cloud для облачных имитационных исследований.	235
Контрольные вопросы к главе V.....	241
Приложения	243

Введение

Рыночная экономика с быстро меняющейся конъюнктурой требует от менеджеров любого звена экономики своевременного принятия адекватных управленческих решений. Информатизация всех сфер национальной экономики, включая аграрно-промышленный комплекс, обусловила необходимость разработки серии учебников и учебных пособий по имитационному моделированию экономических процессов, позволяющему вырабатывать такие решения с использованием компьютерной техники и технологии.

Учебник начинается с описания теоретических основ имитационного моделирования сложных экономических систем, где в доступной форме излагается содержание основных терминов и понятий имитационного моделирования.

Во второй главе описываются основные элементы системы дискретно-событийного моделирования GPSS World. Значительное место в данной части уделено вопросам имитационного проектирования. В эту часть включен раздел «Задания для разработки имитационного проекта в системе GPSS World», который поможет преподавателям дисциплины «Имитационное моделирование» в организации самостоятельной работы студентов.

Третья глава учебника посвящена стремительно развивающемуся последние несколько лет методу имитационного моделирования – методу системной динамики. В этой части приводятся основные элементы системы моделирования Vensim, поддерживающей данный метод, позволяющие пользователю (аналитику) получить основные навыки создания имитационной модели социально-экономической системы и навыки проведения экспериментов с моделью.

Четвертая глава посвящена раскрытию особенностей агентного моделирования. В ней также рассматривается модель, в которой использовано несколько концепций – метод системной динамики, дискретно-событийное моделирование и агентное моделирование.

Пятая глава – новым информационным технологиям в имитационном моделировании.

Труд авторов распределился следующим образом:

Е.В. Худякова, доктор экономических наук, профессор кафедры «Инжиниринг бизнес-процессов» РГАУ-МСХА имени К.А. Тимирязева – главы главы 1- 4 (кроме п. 2.3, 2.4), п. 5.1.

А.А. Липатов – доцент кафедры «Инжиниринг бизнес-процессов» РГАУ-МСХА имени К.А. Тимирязева – главы 2 (п. 2.4), 5 п. 5.2.

Учебник может быть рекомендован студентам, обучающимся по специальности «Прикладная информатика». Оно также может быть рекомендовано для подготовки экономистов и менеджеров, программам «Мастер делового администрирования».

Изучение дисциплины требует знания основ математической статистики и теории вероятностей, информатики и программирования.

ГЛАВА 1. СУЩНОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

1.1 Понятие математического моделирования

Во все времена своего существования человечество стремится все более глубоко постичь окружающий его мир. Одним из методов его постижения является моделирование реальных объектов и систем. Моделирование в процессе изучения окружающего мира представляет собой опосредованное практическое или теоретическое исследование объекта, при котором непосредственно изучается не сам объект, а некоторая вспомогательная искусственная или естественная система. Данная система находится в определенном соответствии с реальной системой и способна замещать его в экспериментах исследователя.

Модель представляет собой абстрактное описание системы (объекта, процесса, проблемы, понятия) в некоторой форме, отличной от формы ее реального существования.

Есть разные виды моделей и моделирования. Среди множества существующих в науке классификаций можно выделить такие виды моделирования:

концептуальное моделирование - представление системы с помощью специальных знаков, символов, операций над ними или с помощью естественных или искусственных языков;

физическое моделирование – моделируемый объект или процесс воспроизводится исходя из физического подобия физических явлений;

структурно-функциональное – представление модели в виде схем (блок-схем), графиков, диаграмм, таблиц и рисунков;

математическое моделирование.

Данное учебное пособие посвящено одному из подвидов математического моделирования. Математическая модель представляет собой выраженное в математических символах совокупность знаний, определяющая представление моделируемого реального объекта. В математической форме выражаются законы, действию которых подчиняется функционирование объекта, а также структурные его составляющие и связи между ними.

Существует множество различных видов математических моделей. Их можно классифицировать, например, следующим образом:

линейные или нелинейные модели;

детерминированные или стохастические;

статические или динамические;

дискретные или непрерывные.

В *линейных* моделях все математические выражения записаны в виде линейных уравнений или неравенств, то есть все переменные представлены в первой степени, что графически изображается в виде прямых линий. *Нелинейные* модели отображают состояние и функционирование системы с помощью как линейных, так и нелинейных взаимосвязей, на графическом изображении модели присутствуют как прямые, так и не прямые линии различной формы.

По характеру выражения причинной обусловленности различают *детерминированные* и *стохастические (вероятностные)* модели. В детерминированных моделях связи носят жесткий, определенный (функциональный) характер. Определенному значению аргумента соответствует строго определенное формулой значение функции. Например, в физической системе сила находится в жесткой определенной зависимости от массы тела $F=am$. В вероятностных системах причинно-следственные связи носят вероятностный характер. Поведение таких систем можно предсказать лишь с определенной степенью вероятности и определенное значение функции при известном значении аргумента может появиться лишь с определенной вероятностью. Например, при увеличении стоимости основных фондов предприятия на 100 тыс руб. объем производства на данном предприятии может принимать с определенной вероятностью разные значения на определенном интервале.

По выражению характера состояния и движения системы различают статические и динамические модели. В *статических системах* переход из одного состояния в другое происходит скачкообразно через определенные интервалы времени. В реальном мире все системы изменяются во времени. Статическими они признаются, если в них не происходит изменений в течение времени данного исследования. В *динамических* системах переход из одного состояния в другое происходит не мгновенно, а постепенно, и данный процесс можно наблюдать и описать.

По характеру отображения динамики системы модели делятся на дискретные и непрерывные. В *дискретных* моделях изменение состояний системы фиксируется через определенные интервалы (отрезки) времени. В *непрерывных* моделях состояние системы фиксируется как в дискретных временных точках, так и в промежутке между ними.

Моделирование имеет следующие цели:

изучение объекта моделирования, когда его структура или поведение не до конца понятно исследователю;

отработка возможных стратегий развития на модели с целью их применения к реальной системе;

прогнозирование поведения системы.

Проведение экспериментов с моделью объекта, а не с самим объектом обусловлено следующими факторами:

это позволяет экономить ресурсы, когда в жизнь воплощается оптимальный вариант структуры системы, найденный с помощью модели;

экономия времени. Эксперимент с реальным объектом для выявления его реакции на какое-либо решение, например, в области макроэкономики зачастую требует нескольких лет и даже десятилетий. В то время как процесс создания модели занимают от нескольких дней до нескольких недель;

возможность реализовать трудновоспроизводимый режим функционирования объекта, например, стихийные бедствия;

многовариантный подход с выбором наилучшего решения из всех возможных;

возможность проследить влияние факторов на результат функционирования системы;

получение более полного представления о реальной системе, выявление ее свойств;

эксперименты на реальной системе могут нарушить установленный порядок ее работы.

1.2. Сущность метода имитационного моделирования

Имитационное моделирование представляет собой частный случай математического моделирования. Имитационное моделирование – это метод, позволяющий строить и реализовывать модели, отражающие процессы, происходящие в системе, во времени. Так как все процессы в реальной жизни происходят на протяжении определенного времени, то метод имитационного моделирования позволяет наиболее адекватным системе образом отразить ее поведение.

Имитационное моделирование производится на основе аналитического математического моделирования. *Аналитические* математические модели предполагают запись основных элементов функционирования системы и внутренних связей в виде алгебраических, интегральных, дифференциальных и иных логических условий. Однако существует класс объектов, для которых по различным причинам не разработаны аналитические модели, либо не разработаны методы решения полученной модели. В этом случае аналитическая модель заменяется имитатором или имитационной моделью. Имитационным моделированием иногда называют получение частных численных решений

сформулированной задачи на основе аналитических решений или с помощью численных методов.

При *имитационном* моделировании логико-математическая модель исследуемой системы представляет собой алгоритм функционирования системы, программно-реализуемый на компьютере. Отличительной чертой имитационного моделирования является воспроизведение процесса функционирования системы во времени и пространстве, т.е. в режиме работы реального объекта.

Комбинированное моделирование сочетает оба подхода.

Развитие компьютерной техники и компьютерных программ породило понятие «компьютерное моделирование». *Компьютерное моделирование* – это метод решения задач анализа и синтеза сложной системы на основе использования ее компьютерной модели. Из описанных выше видов моделирования к нему относят структурно-функциональное и имитационное моделирование.

Классик имитационного моделирования Р. Шеннон приводит такое определение имитационного моделирования: «Имитационное моделирование есть процесс конструирования модели реальной системы и постановка экспериментов на этой модели с целью либо понять поведение системы, либо оценить различные стратегии, обеспечивающие функционирование этой системы».

Метод имитационного моделирования можно представить как экспериментальный метод исследования реальной системы по ее имитационной модели. Имитационная модель отражает процесс функционирования сложной системы, с определенной погрешностью, поэтому имитационное моделирование имеет *экспериментальную природу*. Наиболее объективное представление о работе реальной системы можно получить на основе серии экспериментов, т.е. прогонов модели, так как исход каждого прогона носит случайный характер. Прогон имитационной модели называется *имитационным экспериментом*, в ходе которого компьютер имитирует функционирование системы и вычисляет основные ее характеристики. Имитационный эксперимент представляет собой наблюдение за поведением системы в течение некоторого промежутка времени. Имитационный эксперимент позволяет экспериментировать с моделями систем, которые еще или уже не существуют. Он также дает возможность предсказать поведение существующей системы в будущем или изучить поведение системы в чрезвычайных ситуациях.

Особенностью имитационной модели является то, что она отражает функционирование реальной системы и составляющих ее элементов с сохранением логической структуры системы и **динамики** взаимодействия ее эле-

ментов. То есть описание имитационной модели включает две составляющие:

описание структуры моделируемого объекта и характера взаимосвязей между его элементами;

описание **динамики** функционирования объекта.

При имитационном моделировании логическая структура реальной системы отображается в модели, и при этом имитируется динамика взаимодействий элементов моделируемой системы.

Имитационную модель можно представить в виде нескольких множеств: множества ее элементов (в том числе и внешней среды), множества допустимых связей между элементами, множества рассматриваемых моментов времени. Имитационная модель позволяет воспроизводить моделируемые объекты с сохранением поведенческих свойств (последовательности чередования во времени событий, происходящих в системе).

Имитационная модель – это выполненное на некотором формальном языке (с помощью программы) описание логики функционирования исследуемой системы и взаимодействия отдельных ее элементов во времени, учитывающее наиболее существенные причинно-следственные связи системы с заданием неких начальных значений параметров, соответствующих нулевому моменту времени. Этот метод моделирования не всегда требует составления уравнений, а тем более их решения. При этом взаимосвязи между отдельными элементами системы могут быть представлены в виде аналитических зависимостей.

Тогда объект моделирования представляется в виде *«черного ящика»*, внутренняя структура которого и реакция на изменения внешних воздействий неизвестны. Это бывает, когда моделируемая система является сложной, и исследователю до конца не ясны ее структура и характер происходящих внутри нее процессов. Судить об этом можно по разнице между импульсами и реакциями системы. Иначе – по характеру преобразования информации на входе и выходе из системы. Оказывая на входы системы разнообразные воздействия, исследователь изучает информацию на выходе из системы (Рисунок 1).

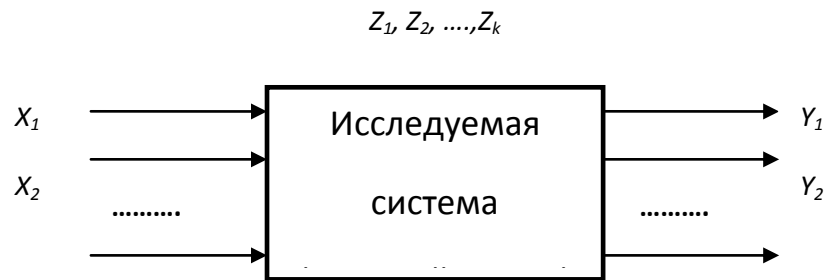


Рисунок 1 – Система, представленная в виде «черного ящика»

Представляемая в виде «черного ящика» система является аналогом или моделью реальной или проектируемой системы. Степень детализации переменных системы может быть различной. Минимальной степенью детализации является тот уровень, когда выделяют один вход и один выход системы. Например, рассматривая зависимость производительности труда от квалификации работника, исследователь абстрагируется от действия таких факторов, оказывающих влияние на результат, как возраст работника, состояние его здоровья, микроклимат в помещении и др.

В этом случае целью моделирования является выработка наиболее приемлемых стратегий управления объектом. Ведь эффективно управлять можно и, не имея представления о внутренней структуре объекта управления (как, например, управляя телевизором не обязательно знать его устройство, а лишь его реакцию на наши внешние воздействия).

Основной целью моделирования является оптимизация параметров системы. В зависимости от цели моделирования используется один из трех подходов:

- исследуется влияние различных факторов на значение выходных характеристик системы;

- нахождение аналитических зависимостей между факторами и результатом;

- поиск оптимального значения параметров системы.

Моделирование экономических процессов представляется более трудной задачей, нежели моделирование физических систем. Это объясняется следующими причинами:

- не все взаимосвязи между элементами, а также выходные характеристики модели поддаются количественному описанию;

- большая роль воздействия на систему случайных факторов;

- существенная подчиненность функционирования системы решениям, принимаемым управляющим ее лицом.

1.3. Представление времени в имитационной модели

Динамика функционирования системы в имитационных моделях реализуется с помощью механизма продвижения модельного времени. В определенные моменты времени моделируемая система приходит в некие **состояния**, описание которых является основным моментом в имитационном моделировании, так как это – *представление динамического поведения системы посредством ее продвижения от одного состояния к другому в соответствии с четко определенными операционными правилами* [4].

При разработке имитационной модели соотносят между собой реальное и модельное время. *Реальное время* – это то, в котором происходит работа самой системы, например оконная рама изготавливается в течение одного часа. *Модельное время* – это время, в масштабе которого организуется работа модели, это виртуальное время, в котором автоматически упорядочиваются все события, причем не обязательно пропорционально реальному времени, в котором развивается моделируемый процесс. События модели происходят в некотором модельном времени. Например, процесс изготовления оконной рамы протекает в модели в течение одной секунды.

Как было отмечено выше, динамика в имитационных моделях реализуется с помощью *механизма продвижения модельного времени*. Модельное время «двигает» процессы внутри модели от одного состояния к другому.

С его помощью:

- в модели синхронизируется течение всех моделируемых процессов;
- моделируемая система переходит из одного состояния в другое;
- производится управление ходом модельного эксперимента.

Выбор механизма изменения модельного времени определяет технологию реализации имитационной модели.

Изменения состояний могут отражаться в модели либо **непрерывно**, либо **дискретно**. В соответствии с этим имитационные модели подразделяются на непрерывные, дискретные и дискретно-непрерывные. В непрерывных моделях величины представляют собой непрерывные **функции времени** (как правило это изменение описывается системами дифференциальных уравнений)(Рисунок 2а). Поэтому пересчет значений переменных величин (продвижение системы во времени) идет при *очень малых* Δt между соседними двумя ее состояниями. В дискретных моделях изменения состояний происходят скачкообразно, и эти изменения называют **событиями** (Рисунок 2б).

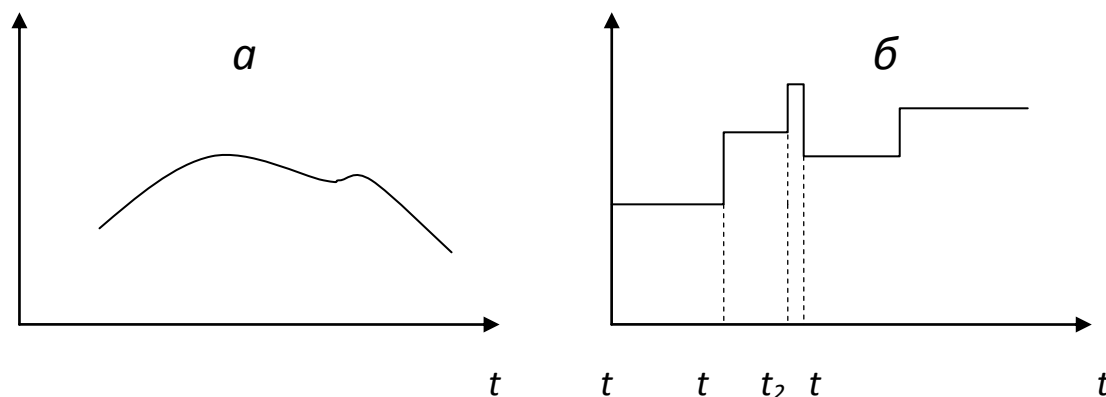


Рисунок 2 – Непрерывные и дискретные функции времени

Указанную на рисунке переменную t называют **модельным временем**. **Механизмы задания модельного времени** встроены в управляющие программы систем имитационного моделирования. Эти программы основаны на одном из двух способов изменения модельного времени – **пошаговом или по-событийном**. При этом вводят глобальную переменную, T , – модельное время. При пошаговом способе применяются фиксированные интервалы изменения модельного времени (все Δt равны между собой). События в модели считаются наступившими в момент окончания каждого такого интервала. Погрешность в измерении временных характеристик системы зависит от величины шага моделирования.

При по-событийном способе применяются переменные интервалы изменения модельного времени, а продвижение модельного времени осуществляется по принципу «от события к событию», от нулевого момента времени к t_1 и т.д. (Δt не всегда равны между собой - Рисунок 1.3б), т.е. модельное время каждый раз меняется на величину, соответствующую интервалу времени до момента наступления очередного события. События обрабатываются в порядке их поступления. Прогнозирование моментов наступления событий не составляет трудностей, если известен закон распределения интервалов между событиями. Если момент наступления события определяется логическим условием, то его необходимо сформулировать и проверять истинность для каждого последующего шага моделирования. Вопрос состоит в том, что понимать под событием. Под этим следует понимать резкое изменение состояния системы. Например, при моделировании работы автозаправочной станции событием является приход очередного автомобиля на заправку.

Пошаговый метод наиболее эффективно применять в тех случаях, когда: события распределены равномерно; сложно предсказать появление от-

дельных событий; событий очень много и они появляются группами. Этот метод менее эффективен с точки зрения использования машинного времени.

Применение по-событийного времени, позволяющего ускорить ход модельного времени, наиболее предпочтительно в случае, если частота наступления событий мала, события распределены неравномерно и необходимо учитывать наличие одновременных событий.

1.4. Классификация имитационных моделей.

Концепции имитационного моделирования

В зависимости от *способов продвижения модельного времени* и характера изменения состояния системы различают непрерывные, дискретные и непрерывно-дискретные имитационные модели. *Непрерывные* имитационные модели предполагают непрерывное изменение переменных, и состояние моделируемой системы меняется как непрерывная функция времени. Это изменение описывается системами дифференциальных уравнений. В *дискретных* имитационных моделях переменные изменяются дискретно (во время наступления очередного события). В *непрерывно-дискретных* моделях совмещаются указанные выше механизмы продвижения модельного времени.

Имитационные модели с точки зрения взаимодействия исследователя с моделью делятся на автоматические и диалоговые. Взаимодействие пользователя с *автоматическими* имитационными моделями сводится к вводу исходной информации и управлению началом и окончанием работы модели. *Диалоговые* модели позволяют управлять ходом моделирования – менять значения параметров по ходу моделирования, приостанавливать его сеанс и т.д.

По *способу изменения модельного времени* имитационные модели бывают с *постоянным шагом* и *переменным шагом*.

По *цели эксперимента* существуют следующие имитационные модели: с оценкой влияния факторов на результирующий показатель, с оценкой взаимного влияния факторов, с оптимизирующим экспериментом.

Концепции имитационного моделирования

В настоящее время известно несколько концепций (подходов) к имитационному моделированию: *метод системной динамики*, *дискретно-событийное моделирование* и *агентное моделирование*.

Первый подход предложен Дж. Форрестером в 60-х годах XX века. Он предполагает описание динамики объекта моделирования без отображения происходящих в объекте элементарных событий. Этот метод относят к методам имитации *потокowego типа*, так как при этом в модели отображаются потоки реальных процессов в виде взаимодействий потоков различной природы (потоков информации, денежных, рабочей силы, материалов, техники).

Второй подход предполагает более детальное описание процессов, происходящих внутри моделируемого объекта. Динамика в данном случае моделируется путем генерирования этих событий в определенной последовательности. Данный метод имитации называют *событийным*. Он используется в основном при описании технологических процессов.

Третий подход является наиболее молодым и наименее разработанным. Математически агентное моделирование оперирует в основном дискретными во времени процессами. Эти модели являются децентрализованными. В отличие от первых двух подходов в них отсутствует централизованный (общий для всех объектов системы) механизм продвижения модельного времени. Исследователь определяет поведение каждого отдельного объекта (агента) системы, а глобальное поведение возникает как результат деятельности многих агентов, действующих по собственным правилам и связанных друг с другом. Поэтому агентные модели называют моделями *«снизу вверх»*.

При моделировании экономических объектов (предприятий, социальных систем, мирового сообщества и др.) наиболее адекватным является потоковая концепция построения моделей, так как их функционирование предполагает не только производственный процесс, круг моделируемых явлений гораздо шире.

При этом предполагается, что основой модели являются *непрерывные* потоки и взаимодействия переменных. Модели должны формулироваться с учетом того, что в них существует непрерывная динамика. Например, производственная мощность предприятия меняется непрерывно, а не дискретно.

1.5. Экспериментальная природа имитационного моделирования

1.5.1. Метод статистических испытаний Монте-Карло

Идею имитационного моделирования о вероятностном его характере, необходимости проведения множества прогонов и статистической оценки результатов прогонов модели можно проиллюстрировать с помощью метода Монте-Карло. Метод получил свое название в результате аналогии с азарт-

ными играми, которыми славится город Монте-Карло. Метод Монте-Карло (*метод статистических испытаний*) широко применяется в статистическом (вероятностном) моделировании, которое является эффективным методом исследования несложных систем (для исследования сложных систем целесообразно применять метод имитационного моделирования). Поскольку результат каждого отдельного испытания носит случайный характер, необходима серия экспериментов для получения *статистической совокупности* (то есть множества) значений какой либо анализируемой величины. Суть метода Монте-Карло состоит в том, что вместо описания случайных явлений аналитическими зависимостями проводится серия испытаний, дающих случайные результаты. Результат конкретного испытания ставится в зависимость от значений случайной величины, распределенной по заданному закону. Проводя серию испытаний, получают статистические данные, которые затем обрабатывают и представляют в виде численных оценок определенной величины.

Метод базируется на предельных теоремах теории вероятности (теорема Чебышева, теорема Бернулли, теорема Пуассона), которые гарантируют высокое качество оценок при большом количестве испытаний (высокое качество оценок означает высокую степень соответствия выборочных данных реальности). Сущность метода можно представить графически.

Вследствие интервального метода задания входных характеристик имитационной модели и наличия генератора случайных чисел, результаты прогона имитационной модели носят случайный характер. Каждая из полученных итоговых характеристик модели является выборочной и не точно, а с определенной погрешностью (ошибкой), отражает реальные характеристики системы. Каждый прогон модели можно интерпретировать как простой эксперимент с моделью. Поэтому можно сказать, что имитационное моделирование имеет экспериментальный (статистический) характер.

Относительно приближенную к реальной (действительной) характеристику модели (системы) можно получить только, проведя серию прогонов и обработав статистическими методами полученную выборочную совокупность. Поэтому имитационное моделирование иногда называют статистическим моделированием.

Сложность в оценке реальных характеристик модели (системы) состоит в том, что характер распределения случайной величины заранее неизвестен, так как производственные процессы находятся под влиянием множества случайных факторов. Поэтому для создания наиболее адекватной модели и получения ее характеристик, наибольшим образом приближенных к реальным, необходимо наблюдать за поведением системы в течение определенного времени.

Для решения определенных выше задач используют метод статистических испытаний Монте-Карло. Алгоритм метода включает выполнение N -испытаний (опытов или прогонов имитационной модели). В каждом испытании задаются случайные значения входных параметров модели x_i , в соответствии с их законом распределения и фиксируются случайные значения параметров y_k (выходных характеристик модели).

Идею имитационного моделирования о вероятностном его характере, необходимости проведения множества прогонов и статистической оценки результатов прогонов модели можно проиллюстрировать с помощью метода Монте-Карло. Метод Монте-Карло (*метод статистических испытаний*) широко применяется в статистическом (вероятностном) моделировании, которое является эффективным методом исследования несложных систем (для исследования сложных систем целесообразно применять метод имитационного моделирования). Поскольку результат каждого отдельного испытания носит случайный характер, необходима серия экспериментов для получения *статистической совокупности* (то есть множества) значений какой либо анализируемой величины. Суть метода Монте-Карло состоит в том, что вместо описания случайных явлений аналитическими зависимостями проводится серия испытаний, дающих случайные результаты. Результат конкретного испытания ставится в зависимость от значений случайной величины, распределенной по заданному закону. Проводя серию испытаний, получают статистические данные, которые затем обрабатывают и представляют в виде численных оценок определенной величины.

Метод базируется на предельных теоремах теории вероятности (теорема Чебышева, теорема Бернулли, теорема Пуассона), которые гарантируют высокое качество оценок при большом количестве испытаний (высокое качество оценок означает высокую степень соответствия выборочных данных реальности).

Общая схема метода Монте-Карло.

Для выявления какой-либо характеристики системы проводят серию испытаний (прогонов модели), в результате которых получают n возможных значений данной величины (X). Эти значения являются случайными, выборочными. Затем из них вычисляют среднее арифметическое значение $\tilde{x} = \frac{\sum x_i}{n}$ и принимают его в качестве оценки (приближённого значения) a^* .

Все выборочные значения удалены на какие-то расстояния от среднего значения и распределены вокруг него согласно какому-либо закону распределения (чаще всего - нормальному). Чем меньше степень рассеяния полу-

ченных значений вокруг среднего, тем меньше средняя ошибка. Полученная выборочная средняя (\tilde{x}) отличается от математического ожидания (a) или реальной (генеральной) средней (\bar{x}) на величину предельной ошибки $\varepsilon = |\tilde{x} - a|$

Если закон распределения случайной величины (X) вокруг среднего – это нормальный закон распределения, то

$$\varepsilon = \frac{t * \delta}{\sqrt{n}}. \quad (1)$$

где δ – среднее квадратическое отклонение выборочных характеристик от среднего значения, t – табличное значение аргумента функции Лапласа.

Метод Монте-Карло требует проведения большого числа испытаний. Теорией метода Монте-Карло разрабатываются способы уменьшения дисперсии используемых случайных величин, в результате чего уменьшается ошибка (ε), допускаемая при замене искомого математического ожидания (a) его оценкой a^* (или (\tilde{x})).

Если же случайная величина (X) распределена по закону, отличному от нормального, то для нахождения предельной ошибки используют формулу (1), где вместо табличного значения функции Лапласа используют табличное значение функции t-распределения Стьюдента.

Например, требуется определить площадь фигуры, алгоритм определения которой неизвестен (Рисунок 3). Однако эту фигуру можно вписать в квадрат и с его помощью сделать это.

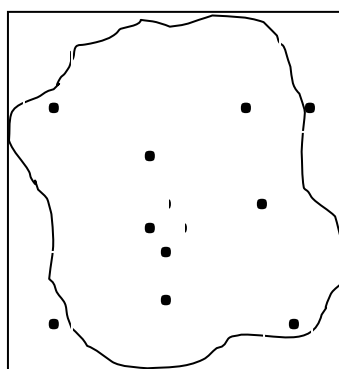


Рисунок 3 – Проведение статистических испытаний

Если «обстрелять» этот квадрат n случайными «выстрелами» (провести некоторое количество испытаний), то часть точек, попавших в него (m), попадет и в фигуру, вписанную в квадрат. По соотношению количества этих точек можно примерно оценить площадь этой фигуры. Если длина стороны

квадрата равна, например, 10 ед., то площадь фигуры определяется по формуле:

$$S_{\text{фиг.}} = S_{\text{квадрата}} \cdot m/n = 100 \cdot m/n$$

На рисунке представлен один прогон. Если провести второй прогон, то расположение точек будет другим, и оцениваемая величина (площадь фигуры) будет несколько иной. Понятно, чем больше «выстрелов» будет сделано, тем точнее будет оценка площади фигуры. В таблице 1 приведены результаты прогонов с разным количеством испытаний. Допустим, нам известно точное значение площади фигуры – 78,54 см². Сравним полученные результаты с этим значением (для каждого количества испытаний проведено по 5 прогонов).

Таблица 1 – Результаты статистических испытаний [2, стр. 36]

Номер прогона	Оценка площади фигуры				
	Число испытаний				
	100	200	1000	5000	10000
1	78,0	79,5	78,8	78,2	78,77
2	70,0	77,0	77,3	78,6	78,23
3	81,0	79,5	80,2	77,7	78,88
4	70,0	77,0	79,5	77,7	78,63
5	79,0	77,0	79,8	79,0	78,21
Средняя	75,6	78,0	79,12	78,26	78,544
Дисперсия	27,3	1,88	1,29	0,3	0,1

На основании полученных результатов можно сделать следующие выводы для любого имитационного эксперимента:

каждый прогон модели при определенном числе испытаний можно рассматривать как один эксперимент на модели;

с увеличением продолжительности прогона (числа испытаний) отклонение измеряемой величины от ее реального значения уменьшается, так как наблюдаемая система переходит в стационарное состояние;

влияние переходных условий можно уменьшить, если увеличить количество экспериментов;

существует предел, за которым увеличение продолжительности прогона модели уже не дает существенного повышения точности результата (см. дисперсию).

Приведенный пример аналогичен идее имитационного моделирования, которое не заканчивается разработкой модели, а требует подготовки и прове-

дения статистического эксперимента. Поэтому результаты имитационного моделирования следует рассматривать как экспериментальные данные, требующие специальной статистической обработки. В частности, возникает необходимость ответить на следующие вопросы:

какова должна быть продолжительность прогона для достижения стационарных условий;

как получить статистически независимые наблюдения;

сколько наблюдений необходимо для обеспечения требуемой точности.

1.5.2. Понятие имитационного эксперимента

Имитационная модель позволяет получить более полное представление о моделируемом объекте для принятия наиболее адекватного решения по управлению данным объектом. Имитационная модель - это модель прогонного (описательного) типа, дающая исследователю возможность сравнения информации на входе в модель и на выходе. По отношению к имитационной модели нельзя оперировать понятием «решение модели». Каждый прогон модели дает результаты, справедливые только для данных значений входных параметров. При изменении параметров модели пользователем очередной прогон модели даст другие результаты. Поэтому наиболее объективное, хотя и приближенное представление о функционировании данной системы можно получить, только осуществив серию прогонов и получив выборочную статистическую совокупность. По данной совокупности можно статистически оценить *доверительный интервал* (интервал, в который с выбранной исследователем вероятностью попадут характеристики реальной или проектируемой системы). «Имитационная модель не способна формировать свое собственное решение в том виде, как это имеет место в аналитических моделях, а может служить в качестве средства для анализа поведения системы в условиях, которые определяются экспериментатором» [4]. В то же время решение имитационное модели может потребоваться в случаях, если:

в модель включены математические выражения, требующие расчетов;

алгоритмы этих расчетов не автоматизированы в системах ЭВМ;

когда алгоритмы модели корректируются в соответствии с имитируемым процессом.

Речь идет о необязательности для имитационных моделей определения **конечного** однозначного решения. Имитационные модели могут служить средством для анализа поведения системы в условиях, определяемых экспериментатором.

Учитывая вышесказанное, а также то, что функционирование большинства систем в реальной жизни в значительной степени зависит от действия случайных факторов, понятно, что выходные переменные имитационной модели (как имеющей стохастический характер) являются случайными величинами. Поэтому, чтобы иметь более объективное представление о функционировании моделируемой системы, проводят большое количество реализаций (прогонов) имитационной модели. Искомый результат определяется как средняя величина по этой массе прогонов, а сама совокупность полученных значений по какому-либо показателю статистически анализируется как **статистическая выборка** с известным из курса статистики алгоритмом ее обработки.

Осуществление серии прогонов модели с различными входными параметрами адекватно проведению экспериментов с моделью. При этом возникает вопрос, сколько именно прогонов модели необходимо провести, чтобы результаты обработки оказались статистически значимыми? С одной стороны, чем больше прогонов, тем точнее выборочные оценки будут воспроизводить реальную систему. С другой стороны слишком большая выборочная совокупность осложняет ее обработку и отрицательно сказывается на точности исследования вследствие большой вероятности появления ошибок вычисления, а также увеличивает затраты машинного (компьютерного) времени на имитацию. Для решения этих проблем существует процедура *планирования имитационного эксперимента*. Выделяют два вида планирования имитационного эксперимента. Первый – *стратегическое планирование*, заключающееся в разработке такого плана эксперимента, который позволяет определить характер взаимосвязи между вводимыми характеристиками модели и ее откликами (выходными характеристиками), а также оптимизировать величину управляемых параметров, позволяющих получить максимум или минимум результирующих характеристик модели. Второй – *тактическое планирование*. Это – планирование технических приемов проведения прогонов – определение длительности прогона, оценка точности результатов моделирования и т.д.

В целом планирование эксперимента включает:

- подготовку входных данных для каждого прогона модели (эксперимента);

- определение количества прогонов модели;

- определение длительности каждого прогона модели;

- выявление длительности переходного процесса моделирования (до момента перехода системы в стационарное состояние, когда выходные характеристики наиболее адекватны функционированию реальной системы);

разработку стратегий сбора данных для каждого прогона;
методику оценки точности и чувствительности результатов моделирования;
планирование возможных сценариев развития системы, которые требуется оценить с помощью модели;
оптимизацию целевой функции (если это требуется по условиям задачи).

Имитационные эксперименты с моделью называют *направленными вычислительными экспериментами*. Наиболее часто используют такие типы направленного вычислительного эксперимента, как^{*}:

1) Оценка существенности влияния какого-либо фактора (или факторов) на выходные характеристики модели – дисперсионный анализ (для заведомого включения в рассматриваемый круг факторов наиболее значимых, проводится *отсеивающий* несущественные факторы эксперимент).

2) *Оптимизирующий* параметры системы эксперимент.

В своем функционировании реальные системы испытывают влияние случайных факторов, которые могут быть отражены в модели. Например, интервал времени между приездом автомобилей на заправочную станцию – это случайная величина. Или частота поломок трактора на посевных работах – также величина случайная. Такие факторы могут быть отражены в модели как случайные функции, события и величины с помощью **генераторов случайных чисел (датчиков случайных чисел)**, которые входят в состав всех современных систем имитационного моделирования. Генерируемые числа равномерно распределены на интервале $[0;1]$.

Соответствие построения имитационных моделей работе реальных систем возможно благодаря использованию генератора случайных чисел.

1.6. Этапы имитационного моделирования

Процесс имитационного моделирования включает несколько этапов:

Изучение реальной системы. Понимание сущности проходящих в ней процессов и ее структуры.

Формулировка целей и задач моделирования. Составление списка входных переменных (факторов), вспомогательных переменных (рассчитываемых в модели) и уровней, в том числе тех, которые отражают цели моделирования и цели функционирования системы (результативные показатели).

^{*} Более подробно о типах экспериментов - в [4, 5].

Разработка концептуальной модели, предполагающая отражение с помощью визуально понятных и читаемых абстрактных изображений, описывающее логику функционирования системы и характер моделируемого управляющего воздействия на нее со стороны лица, принимающего решения (или разработчика модели).

Реализация модели с помощью какой-либо компьютерной программы (возможно предполагающей графический интерфейс и отражение анимации элементов модели).

Оценка адекватности модели – то есть ее соответствия реальной системе (например, если моделируемое предприятие имеет прибыль в данном году 2 млн. руб., а по результатам реализации модели получилось 10 млн руб., то модель неадекватна).

Калибровка модели – процесс соотнесения математической модели камеры с параметрами функционирования конкретной системы.

Планирование и проведение компьютерного эксперимента.

1.7. Области применения имитационного моделирования

Имитационное моделирование успешно используется при моделировании сложных систем, когда функциональные зависимости могут описываться громоздкими математическими соотношениями. Некоторые взаимосвязи в имитационном моделировании описать аналитически (с помощью формул, зависимостей) невозможно (или возможно, но уравнения, например некоторые дифференциальные, настолько сложны, что алгоритм их решения в настоящее время математической науке неизвестен).

Поскольку имитационное моделирование относится к динамическому, то его используют, когда существует необходимость наблюдения за поведением объекта в течение определенного времени. Имитационная модель имеет практическую ценность тогда, когда наибольшим образом отражает те свойства, которые влияют на выбранный критерий эффективности (минимум себестоимости производимой на предприятии продукции, максимум чистого дохода и др.).

Имитационное моделирование используется в самых разных областях. В экономике имитационные модели могут быть использованы для моделирования производственных систем, в логистике, маркетинге, моделировании бизнес-процессов. В социально-экономических исследованиях – при моделировании экономических реформ, в политологии, социологии и др. Имитаци-

онное моделирование используется также при моделировании глобальных мировых процессов.

Имитационное моделирование применяется также при исследовании стохастических систем, когда исследуемая система может быть подвергнута влиянию случайных факторов. Оно позволяет проводить исследования систем при частичном отсутствии информации о них. Имитационное моделирование позволяет также исследовать большое количество альтернатив с целью принятия оптимального управленческого решения. При этом оно дает возможность найти ответ на вопрос «Что будет, если?...».

Учитывая принципиальные отличия аналитических и имитационных моделей, наиболее эффективно применение имитационных моделей в следующих случаях:

- если не существует законченной постановки задачи и идет процесс познания объекта;

- когда описание процесса или отдельных его элементов невозможно с помощью аналитических формул;

- когда за поведением системы необходимо наблюдать в течение определенного времени;

- при изучении новой, ранее не наблюдавшейся ситуации;

- если исследуемая система является элементом более сложной системы;

- при подготовке специалистов и освоении новой техники в качестве тренинга.

Как отмечает Р. Шеннон, метод имитационного моделирования для предприятия привлекателен благодаря своей простоте. Но в то же время существуют факторы, препятствующие использованию метода имитационного моделирования в экономических исследованиях. Это следующие факторы:

- 1) отсутствие на предприятии высококвалифицированных программистов;

- 2) значительное время для создания модели;

- 3) модель не всегда адекватна реальной системе. В этом случае использование ее для выработки конкретных управленческих решений некорректно;

- 4) имитационные модели относятся к вероятностным, поэтому требуется специальная процедура оценки чувствительности модели к изменению каких-либо входных параметров;

- 5) Вероятностный характер полученных результатов моделирования и вызванная этим необходимость проведения нескольких прогонов модели (экспериментов).

1.8. Программные средства создания имитационных моделей

В настоящее время для реализации имитационных моделей существует множество универсальных языков моделирования – UML (Universal Modeling Language). Для облегчения работы с программой-имитатором, в которую для реализации модели преобразуется ее формальное описание, используются специальные языки и системы имитационного моделирования (Рисунок 4).

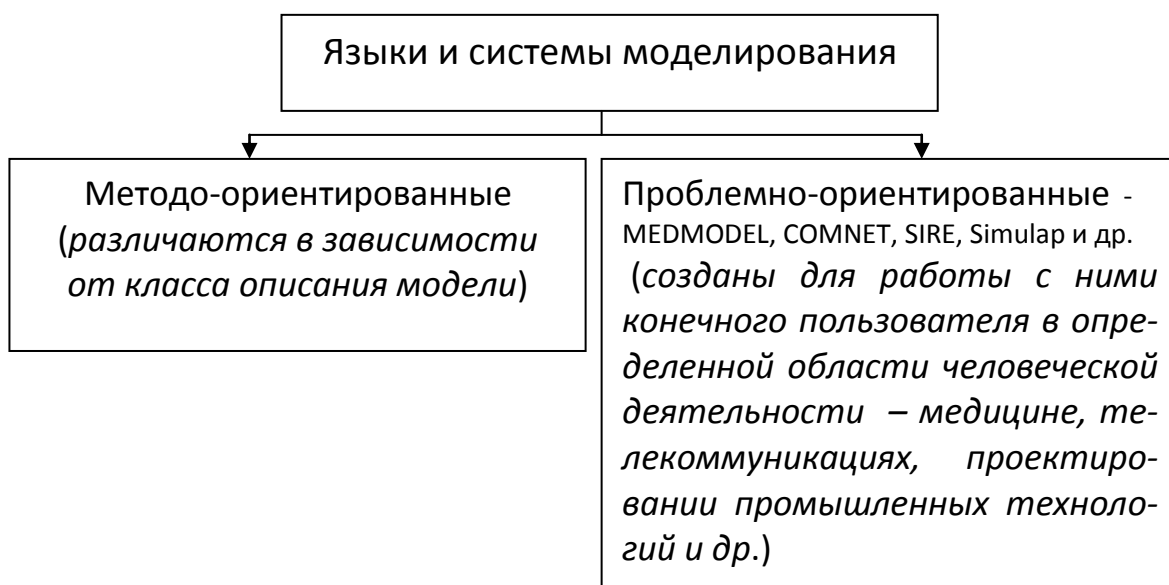


Рисунок 4 – Виды языков и систем имитационного моделирования

Система моделирования – это более широкое понятие, нежели язык моделирования. Она включает, кроме языка моделирования, *язык управления системой моделирования* (команд интерактивного взаимодействия с пользователем) и *управляющую программу* (программное средство, обеспечивающее трансляцию модели, генератор случайных чисел, инструменты сбора статистической информации и др.).

Системы моделирования различаются:

- по базовой схеме алгоритмизации;
- по альтернативной схеме алгоритмизации;
- по синтаксической основе языка моделирования;
- по средствам проблемной ориентации в языке моделирования.

Базовая схема алгоритмизации – это совокупность понятий, используемых для алгоритмизации моделируемой системы, представленных в языке моделирования.

В некоторых системах моделирования присутствуют *альтернативные схемы алгоритмизации*. Так, например, в системы Powersim и Ithink, поддерживающие класс непрерывного моделирования, встроен аппарат непрерывного моделирования. В системах Extend и ProcessModel реализован алгоритм дискретного моделирования. А такая система моделирования как AnyLogic, совмещает транзактно-ориентированный способ описания дискретных систем, динамическое моделирование, диаграммы состояний универсального языка моделирования UML и агентный подход в моделировании.

Языки моделирования имеют специфическую для каждого *синтаксическую основу*. По синтаксической основе различают языки, вложенные в базовый язык моделирования или расширяющие его, а также языки с жестким синтаксисом.

Для более облегченного взаимодействия пользователя с системой моделирования при моделировании определенной проблемной области в известные языки моделирования могут вводиться специальные *средства проблемной ориентации* – макросы, подпрограммы и процедуры.

Среди множества компьютерных технологий имитационного моделирования исследователь должен выбрать наиболее подходящий язык для данного класса моделируемых объектов. Выделяют три класса моделируемых объектов – дискретные, непрерывные и дискретно-непрерывные, и соответственно этому – три класса языков моделирования (Рисунок 5).

Какими факторами руководствуются при выборе системы моделирования? Непрерывные системы моделирования, поддерживающие методы системной динамики, используются в тех случаях, когда:

- материальные объекты движутся;
- в моделируемой системе присутствуют непрерывные информационные потоки;
- в системе можно выделить множество связанных материальных объектов.

Системы дискретного типа используются, если:

- в системе наблюдается дискретный характер ее изменения;
- изменения происходят неравномерно и связаны с появлением событий;
- по системе перемещаются единичные объекты.

Дискретно-непрерывное моделирование используется, если система обладает совокупностью свойств первой и второй групп.

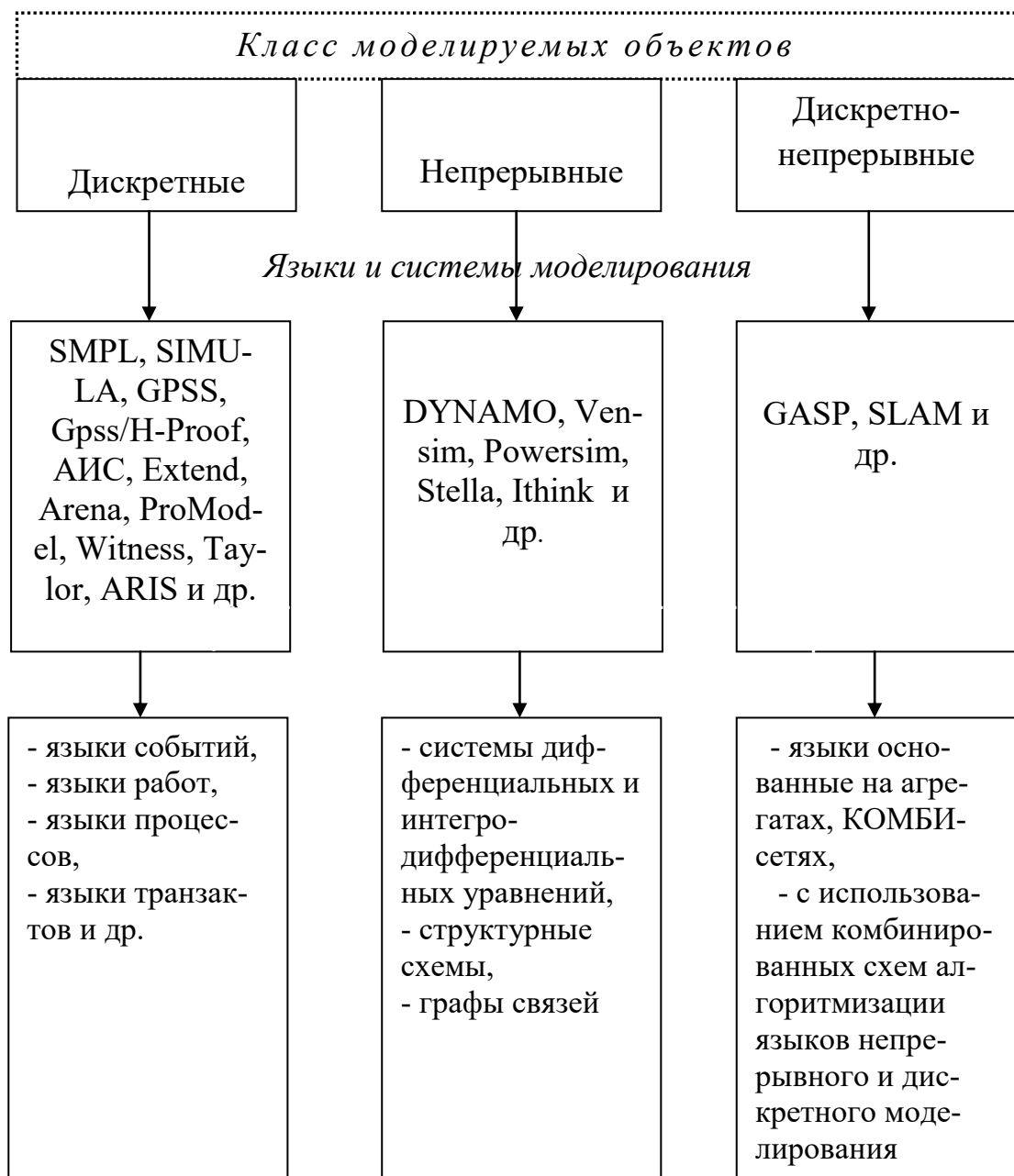


Рисунок 5 – Классы объектов и языки моделирования

Контрольные вопросы к главе I

1. Что представляет собой моделирование вообще? Математическое моделирование?
2. Какие виды моделирования вам известны?
3. Что такое математическая модель и математическое моделирование? Для каких целей осуществляется имитационное моделирование?

Чем отличается имитационное моделирование от аналитического? Понятие имитационного моделирования. Что такое имитационная модель?

Принцип «черного ящика в имитационном моделировании».

Каковы цели имитационного моделирования?

Что представляет собой модельное время?

Как классифицируются имитационные модели в зависимости от способа продвижения модельного времени?

Классификация имитационных моделей.

Концепции имитационного моделирования.

Метод статистических испытаний Монте-Карло.

Понятие имитационного эксперимента.

Каковы этапы имитационного моделирования?

Назовите области применения имитационного моделирования.

Программные средства автоматизации имитационного моделирования.

ГЛАВА 2. ДИСКРЕТНО-СОБЫТИЙНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В АПК

Система моделирования GPSS World поддерживает дискретно-событийный подход к имитационному моделированию. Данную систему (как и примерно около ста подобных систем, существующих сегодня на рынке) можно *рассматривать как глобальную схему обработки заявок со стохастическими элементами* (то есть систему массового обслуживания).

2.1. Системы массового обслуживания и их сети

Система массового обслуживания представляет собой некий объект, состоящий из обслуживающих устройств, логически связанных между собой. Например, обслуживание покупателей в магазине можно классифицировать как систему массового обслуживания. Здесь имеются все классические элементы СМО: *очередь, обслуживающее устройство (продавец), входящий и выходящий поток заявок на обслуживание (покупатели)* (Рисунок 3.1).

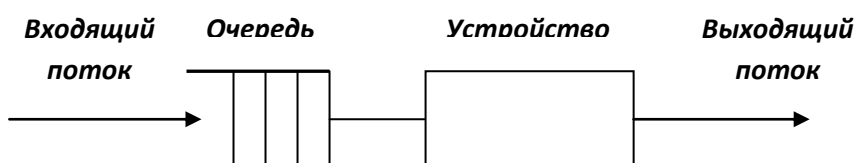


Рисунок 6 – Схема простейшей системы массового обслуживания

Любая СМО имеет такие характеристики, как:
закон поступления заявок на обслуживание;
дисциплина выхода из очереди;
правило обслуживания;
режим работы СМО.

Закон поступления заявок на обслуживание может быть *детерминированным* (например, одна заявка каждые 3 мин) или *вероятностным* (требования могут появляться с равной вероятностью (равномерно) в интервале 10 ± 2 мин). Входящий поток требований описывается распределением вероятностей интервалов времени между соседними требованиями.

Пример. Построить вариационный ряд распределения вероятностей временных интервалов поступления заявок в СМО.

Порядко-	1	2	3	4	5	6	7	8	9	10	11
----------	---	---	---	---	---	---	---	---	---	----	----

вый номер заявки											
Интервал времени после поступления предыдущей заявки, мин	1	1	2	1	1	4	3	3	2	1	2

Вариационный ряд распределения вероятностей интервалов времени между соседними требованиями будет следующим (Рисунок 2.2):

Интервал времени после поступления предыдущей заявки, мин	1	2	3	4
Количество интервалов указанной длины	5	3	2	1
Вероятность соответствующей длины интервала	0,455 (5/11)	0,273 (3/11)	0,182 (2/11)	0,09 (1/11)

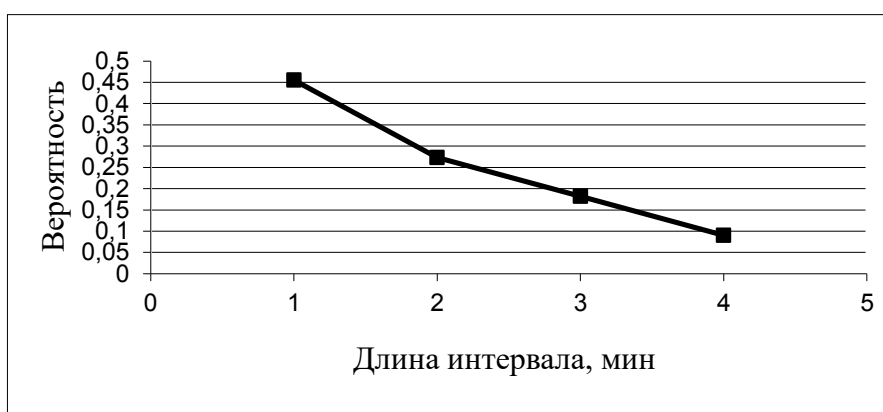


Рисунок 7 – Полигон распределения вероятностей длины временных интервалов между соседними требованиями

Поскольку в теории СМО предполагается, что требования поступают от большого числа источников за определенный интервал времени, входящий поток требований имеет распределение Пуассона, которое задается экспоненциальным законом распределения.

Среднее количество требований в СМО \bar{N} определяется по формуле:

$$\bar{N} = \lambda T,$$

где λ – интенсивность поступления заявок (количество заявок в единицу времени); T – среднее время пребывания требования в системе.

Дисциплина выхода из очереди может быть одной из следующих:
случайная;
«раньше пришел – раньше обслужился»;
«последний пришел – первый обслужился»;
по параметрам (например, ветераны войны обслуживаются вне очереди).

Правила обслуживания включают количество устройств обслуживания, количество одновременно обслуживаемых требований и дисциплину обслуживания. СМО предполагают наличие *одноканальных* или *многоканальных* устройств. Например, заправка осуществляется одной бензиновой колонкой или несколькими. Обслуживание может быть *однофазным* (например, продавец в магазине сам получает деньги и отпускает товар) или *многофазным* (сначала покупатели уплачивают деньги в кассе, получая чек, а затем идут к продавцу за товаром).

Дисциплины обслуживания могут быть *бесприоритетными* и *приоритетными*. Приоритеты могут быть относительными и абсолютными. При относительном приоритете поступление заявки не прерывает текущего обслуживания заявки. При абсолютном приоритете поступление такой заявки прерывает текущее обслуживание другой заявки.

Режим работы предполагает, во-первых, учет наличия или отсутствия вероятности поломки обслуживающего устройства, во-вторых, – замедление или прекращение обслуживания, например, в связи с временным отсутствием деталей для их обработки на токарном станке.

Основные характеристики работы СМО

1. Среднее количество требований (заявок) в очереди (\overline{N}_q):

$$\overline{N}_q = \lambda w,$$

где λ – интенсивность поступления заявок; w – среднее время пребывания требований в очереди.

2. Среднее количество требований в устройстве (\overline{N}_s):

$$\overline{N}_s = \lambda \bar{x},$$

где \bar{x} – среднее время обслуживания в устройстве.

3. Коэффициент загрузки (ρ) определяет, какую часть времени устройство было занято на протяжении времени наблюдения за работой СМО.

4. Вероятность обслуживания требования ($P_{об}$):

$$P_{об} = \frac{N_{об}}{N},$$

где $N_{об}$ и N – соответственно количество обслуженных требований и общее количество требований, поступивших в СМО за время наблюдения.

5. Вероятность отказа в обслуживании ($P_{отк}$):

$$P_{отк} = \frac{N_{отк}}{N},$$

где $N_{отк}$ – количество требований, которым отказано в обслуживании.

6. Среднее время пребывания в очереди ($t_{оч}$):

$$t_{оч} = \frac{\sum_{i=1}^k t_i^{оч}}{N_{об}},$$

где $t_i^{оч}$ – время пребывания i -го требования в очереди.

7. Среднее время обслуживания в устройствах (при многоканальных СМО) ($t_{об}$):

$$t_{об} = \frac{\sum_{i=1}^k t_i^{об}}{N_{об}},$$

где $t_i^{об}$ – время обслуживания i -го требования.

8. Общее среднее время пребывания требования в СМО:

$$T = t_{об} + t_{оч}$$

Сети систем массового обслуживания

Сеть СМО представляет собой совокупность одноканальных и многоканальных СМО, связанных между собой каналами передачи требований (заявок). Существуют разомкнутые и замкнутые сети СМО. Примером разомкнутой сети может являться цех по обработке куриных тушек, куда извне поступают куры и откуда выходят, пройдя несколько фаз обработки, куриные тушки. Транзактами при этом являются тушки. Примером замкнутой сети может являться система, включающая ремонтную бригаду и трактора, ко-

торые она ремонтирует. Трактора, периодически выходя из строя, попадают в бригаду ремонтников, которые, отремонтировав технику, снова возвращают ее в поле. При очередной поломке техника снова попадает в ремонтную мастерскую и т.д. Устройство обслуживания (одноканальное или многоканальное) в теории сетей СМО называется **узлом**.

Представление динамического объекта в виде сети СМО и расчет основных характеристик функционирования сети проводят с целью выявления в ней «узких мест». Узким местом в сети называют тот узел, коэффициент загрузки которого близок к единице. Понятно, что именно такие узлы обуславливают производительность работы всей сети, тормозя работу, так как, вероятно, именно там образуется простой (очередь) заявок. Найдя узкое место, можно принять такие управленческие решения, как, например, замена оборудования данного узла на более производительное или введение дополнительных каналов обслуживания в данный узел и др.

При анализе сетей СМО исходят из *гипотезы*, что количество требований, поступивших в некий узел за время T равно количеству требований, покинувших этот узел. Наиболее часто при анализе сетей СМО рассчитывают и анализируют такие их характеристики как:

1. Коэффициент использования узла (U_k):

$$U_k = \frac{B_k}{T},$$

где B_k – общее время обслуживания узлом k .

T – общее время наблюдения за системой или время моделирования.

2. Среднее время обслуживания в узле k (S_k):

$$S_k = \frac{B_k}{C_k},$$

где C_k – общее количество требований, прошедших через сеть.

3. Относительная частота перехода требований между узлами k и j :

$$q_{kj} = \frac{C_{kj}}{C_k}$$

или, если требование поступает из внешней среды в узел j

$$q_{kj} = \frac{A_{0j}}{A_j},$$

где C_{kj} – количество требований, покинувших узел k и поступивших в узел j ;

A_{0j} – количество требований, поступивших в узел j из внешней среды;

A_j – количество требований, поступивших в узел j .

Для замкнутой сети $A_0 = C_0$.

4. Производительность узла (X_k):

$$X_k = \frac{U_k}{S_k}.$$

2.2. Основные элементы GPSS World

Система GPSS (General Purpose Simulating System) относится к объектно-ориентированным языкам моделирования. Она была разработана в 1961 г. и является предшественником таких современных систем имитационного моделирования как Extend, Arena, Process Model, Taylor, WITNESS и др. Изучение инструментов имитационного моделирования целесообразно начинать именно с GPSS, так как она имеет четкую блочно-ориентированную структуру. Концепция GPSS разработана с ориентацией на моделирование работы систем массового обслуживания (СМО). В GPSS есть средства, являющиеся аналогами систем массового обслуживания – заявки (транзакты), обслуживаемые устройства, очереди. В то же время GPSS позволяет моделировать не только СМО.

Несмотря на применяемый в GPSS аппарат СМО данную систему нельзя рассматривать как компьютерный алгоритм аналитического моделирования СМО. GPSS – это язык имитационного моделирования. В отличие от аналитического моделирования СМО ее имитационная модель, реализуемая с помощью GPSS, может отражать, например, различные законы распределения (то есть не только равномерное) поступления, обслуживания заявок или выбора их из очереди. Имитационная модель позволяет исследовать поведение системы не только в стационарном режиме. Каждый прогон модели представляет собой одну случайную реализацию процессов обслуживания в СМО. Поэтому для получения более объективных данных о функционировании системы необходима серия случайных прогонов, а затем статистическая обработка результатов испытаний.

GPSS относится к системам *дискретного* типа, то есть моделирует системы с дискретными событиями. Это – система относится к классу *транзактно-ориентированных* систем моделирования.

Основными элементами языка GPSS являются блоки и транзакты. **Транзактами** называются динамические элементы системы. Например, при моделировании работы тракторной ремонтной мастерской транзактами являются трактора, а при моделировании работы автозаправочной станции – автомобили, приходящие на заправку. Транзакты имеют свойства, отражаемые в нумеруемых параметрах транзактов. Традиционно транзакты имеют такие параметры, как порядковый номер, номер блока, в котором транзакт находится в данный момент, номер следующего блока, приоритет транзакта и др. Транзактам может быть присвоен и специфический набор параметров. Так, например, транзакты – яблоки на сортировке имеют такие специфические параметры, как цвет, размер, наличие или отсутствие гнилых мест и т.д.

Последовательность блоков модели показывает направление, в котором перемещаются транзакты, поступающие в модель с частотой, задаваемой исследователем. **Блоки GPSS** содержат набор параметров для обращения к подпрограммам, написанным на макроассемблере или на языке Си. В каждом блоке транзакт задерживается на какое-то время. После того, как транзакты пройдут все блоки, они должны быть выведены из модели. Каждый транзакт появляется в системе в определенный момент времени. Как было отмечено выше, поток заявок можно отобразить некоторым законом их поступления (равномерное, экспоненциальное, нормальное и др.). Транзакты задерживаются в обслуживающих устройствах, образуя очереди.

Обработка транзактов происходит в так называемых **устройствах**. Например, в парикмахерской под устройством принимается парикмахер.

Для моделирования объектов, обладающих некой емкостью, в GPSS существует **память**, которая может обслуживать сразу несколько транзактов.

Для сбора статистических данных в GPSS существуют статистические объекты **очереди** (собирают такую информацию о состоянии очереди, как ее средняя длина, среднее время пребывания в ней транзакта и др.) и **таблицы** (собирают статистику для выявления характера статистических распределений и построения гистограмм).

Рекомендуем начать изучение данного языка по имеющимся в настоящее время на рынке учебникам – [5, 6 и др.]. Здесь приведем лишь некоторые, наиболее важные моменты.

Язык GPSS включает следующие объекты (Рисунок 2.3). Каждый объект имеет присущие ему свойства, называемые **стандартными числовыми атрибутами (СЧА)**.

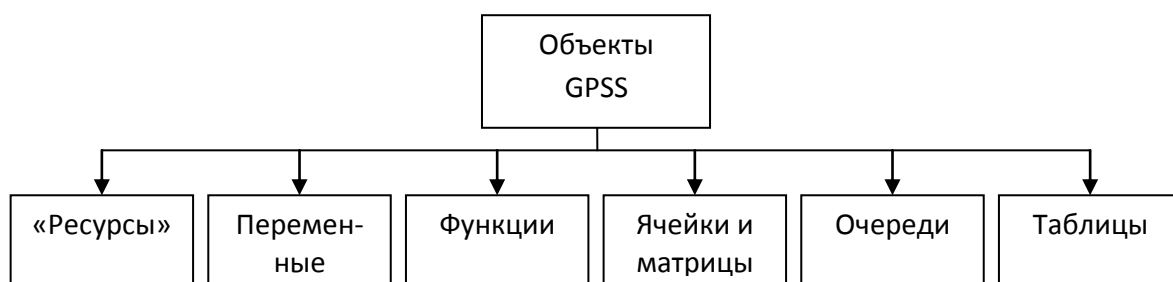


Рисунок 8 – Классификация объектов GPSS

К **объектам типа «ресурс»** относятся одноканальные и многоканальные устройства, а также логические ключи. Многоканальные устройства – это несколько параллельно работающих одинаковых устройств. Логические ключи блокируют или изменяют направление движения транзактов.

Переменные в GPSS – двух видов – арифметические и булевы. **Арифметические переменные** позволяют вычислять арифметические выражения, а **булевы переменные**, включающие логические операторы, позволяют пользователю проверять несколько условий и изменять направления движения транзактов.

В GPSS World можно задавать непрерывные и дискретные **функции** с помощью операторов описания функций.

Для хранения числовой информации, которая может считываться любым транзактом, используются **ячейки и матрицы сохраняемых величин**.

В случае если обслуживающее устройство занято, транзакты встают в **очередь** (например, клиенты в банке в ожидании обслуживания). Могут быть заданы приоритеты выхода из очереди.

Таблицы предназначены для сбора статистики о случайных величинах.

В GPSS встроены специальные средства, являющиеся аналогами систем массового обслуживания – обслуживающие устройства, очереди, транзакты (заявки). То есть алгоритм СМО служит как бы «внутренним двигателем» GPSS. Но в то же время GPSS является гибкой системой, способной моделировать не только СМО, но и, например, распределение ресурсов, работу склада, банка и т.д.

2.3. Типы операторов GPSS World

Операторы GPSS делятся на три типа:

- 1) блоки;
- 2) операторы описания данных;
- 3) команды.

Понятие блока было дано выше. Формат блоков в GPSS World следующий:

[<Метка>] <Операция> <Операнды> [<Комментарии>]*

Метка должна начинаться с буквы (все буквы в модели – латинские). Чтобы название метки (устройства, очереди, функции, переменной и т.д.) интерпретатор не спутал с названием операции, целесообразно использовать символ подчеркивания. Например, обслуживающее устройство – «кладовщик» может быть в модели названо «K_ladovsik».

Операция – это непосредственно команда, которая должна быть выполнена при входе транзакта в данный блок. Для написания каждой операции в GPSS World существует четко установленная грамматика.

Операнды задают информацию, специфичную для данного блока. Их не может быть больше семи: A, B, C, D, E, F, G. Операнды отделяются запятыми. Если отдельный операнд опускается, то на его месте ставится пробел.

Комментарии отделяются от поля операндов символом «;» или «*». Они могут быть написаны на русском языке и служат для удобства ориентирования в модели.

Команды GPSS World используются как для построения модели (то есть они могут содержаться в тексте модели), так и для интерактивного взаимодействия с моделью (например, команда **START**, речь о которой пойдет ниже).

Поскольку система GPSS World достаточно хорошо описана в [5, 6, 7], форматы основных блоков GPSS World приведем здесь справочно.

Вход и выход транзактов из модели

I. Блок GENERATE (генерировать) – введение транзактов в модель.

Формат блока:

GENERATE [A],[B],[C],[D],[E]

Операнд	Значение операнда	Значение по умолчанию
A	Средний интервал времени (это может быть число, СЧА, встроенная функция)	0
B	Половина поля допуска равномерно распределенного интервала (число, СЧА)	0

* Здесь и далее в описании форматов блоков GPSS World скобки [] означают, что эта часть формата блока может быть опущена.

C	Смещение интервалов	Смещение отсутствует
D	Ограничитель транзактов	∞
E	Уровень приоритета транзактов	0

Примеры.

1. Транзакт входит в модель через каждые 10 ± 3 минуты:

GENERATE 10,3

2. Транзакт входит в модель через каждые 8 минут:

GENERATE 8

3. Приход в модель через каждые 30 ± 4 минуты 5 транзактов:

GENERATE 30,4,,5

4. Приход в нулевой момент времени трех транзактов с приоритетом 2:

GENERATE ,,5,2

5. Приход в модель через 20 ± 7 минут транзактов с приоритетом, более высоким, чем в примере №4, причем первый транзакт приходит через 10 мин после начала работы системы:

GENERATE 20,7,10,,3.

II. Блок **TERMINATE** (завершить) – удаление транзактов из модели.

Формат блока:

TERMINATE [A]

Операнд **A** – величина уменьшения *счетчика завершения*. По умолчанию $A = 0$.

При попадании в этот блок транзакты уменьшают величину счетчика, включаемого командой **START** (начать) на величину **A**.

Примеры:

1. Закончить моделирование, когда через модель пройдет 500 транзактов:

GENERATE 20,7

...

TERMINATE 1

START 500

2. Ввести в модель счетчик времени (таймер-сегмент), по истечении которого (например, 300 мин) процесс моделирования остановится:

GENERATE 300

TERMINATE 1

(здесь генерируется 300 минут, которые убывают по одной минуте).

Таймер сегмент присутствует в большинстве моделей и вводится в конце основного сегмента модели. при этом основной сегмент заканчивается оператором **TERMINATE** (заметим, что значение операнда **A** использовано по умолчанию, то есть $A = 0$).

Занятие и освобождение обслуживающего устройства

I. Блоки **SEIZE** (занять) – занятие обслуживающего устройства и

II. **RELEASE** (освободить) – освобождение обслуживающего устройства.

Формат блока **SEIZE**:

SEIZE A

Операнд	Значение операнда	Значение по умолчанию
A	Имя (символьное или числовое) занимаемого устройства	Ошибка

Формат блока **RELEASE**:

RELEASE A

Операнд	Значение операнда	Значение по умолчанию
A	Имя (символьное или числовое) освобождаемого устройства	Ошибка

Примеры:

1. Автомобиль занял бензиновую колонку (название колонки – Kol):

SEIZE Kol

2. Автомобиль освободил бензиновую колонку:

RELEASE Kol

Обслуживание в устройстве

Блок **ADVANCE**

Формат блока:

ADVANCE A,[B]

Операнд	Значение операнда	Значение по умолчанию
A	Средне время задержки на обслуживание (число, СЧА)	0
B	Половина поля допуска равномерно распределенного времени задержки (число, СЧА)	0

Примеры:

1. Клиент в банке обслуживается в среднем за 6 ± 2 минуты:

ADVANCE 6,2

2. Автомобиль занимает колонку Kol и заправляется в течение 2 минут, затем освобождает колонку (составить сегмент модели):

SEIZE Kol

.....

ADVANCE 2

RELEASE Kol

Моделирование входа и выхода из очереди

Блок QUEUE (встать в очередь) и Блок DEPART. В модели регистрируется и потом выдается в стандартном отчете о результатах моделирования статистика об очереди перед обслуживающим устройством (средняя длина очереди, время пребывания транзакта в очереди и т.д.). После занятия обслуживающего устройства транзакт должен покинуть очередь. Это отображается блоком **DEPART**.

Форматы блоков:

QUEUE A,[B]

DEPART A,[B]

Операнд	Значение операнда	Значение по умолчанию
A	Имя очереди, в которую необходимо стать (покинуть) транзакту	Ошибка
B	Число единиц, на которое увеличивается (уменьшается) длина очереди (число, СЧА)	1

Очереди в GPSS имеют несколько стандартных числовых атрибутов (СЧА), после которого записывается имя очереди. Так, длина очереди имеет СЧА **Q**, а среднее время пребывания транзактов в очереди – **QT**. Если имя – символьное, а не числовое, то чтобы отделить СЧА от имени очереди используется знак \$. При выходе транзакта из очереди через блок **DEPART** транзакт может уменьшать длину очереди на другое количество единиц, чем он увеличивал длину очереди в блоке **QUEUE**.

Примеры:

1. Увеличить длину очереди Ochered на три единицы:

QUEUE Ochered,3

2. Прибыв в банк, клиент встает в очередь (имя очереди Ochered). После того, как подойдет его очередь на обслуживание работником банка (имеющего имя Bankir), клиент, подходя к банкиру, покидает очередь; расчеты с клиентом длятся 5 ± 2 минуты:

QUEUE Ochered
SEIZE Bankir
DEPART Ochered
ADVANCE 5,2
RELEASE Bankir

Обозначения стандартный отчет со статистикой очереди означают следующее:

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE.TIME	AVE.(–0)	RE- TRY
Назва- ние оче- реди (сим- вольное или чис- ловое)	Макси- мальная длина очереди	Текущая длина очереди	Общее кол–во входов	Кол–во нулевых входов	Средняя длина очереди	Среднее время пребыва- ния транзакта в очереди	Среднее время пребыва- ния транзакта в очереди без учета нулевых входов	Кол–во транзактов, ожидающих специальных условий

Знания приведенных выше блоков достаточно для построения простой имитационной модели.

Пример. Смоделировать работу цеха по стрижке овец в течение рабочей смены (480 мин.), если овцы поступают в цех равномерно каждые 5 ± 2 мин. В цехе работает один работник, который стрижет одну овцу в среднем за 7 ± 3 минуты.

GENERATE 5,2	;поступление овец на стрижку
QUEUE Ochered	;вход овцы очередь на стрижку
SEIZE Striwnka	;занятие места у мастера
DEPART Ochered	;выход из очереди на стрижку
ADVANCE	7,3 ;стрижка овцы
RELEASE	Striwnka ;освобождение мастера
TERMINATE	;выход овцы из цеха

GENERATE 480 ;моделирование работы цеха в
;течение смены
TERMINATE 1 ;убывание времени по 1 минуте
START 1

Переход транзакта в блок,
отличный от последующего

Блок TRANSFER (передать) – переводит транзакты в блоки, отличные от последующего. Данный блок имеет девять режимов. Приведем здесь некоторые из них.

а). Режим безусловной передачи.

Формат блока:

TRANSFER ,B

Операнд	Значение операнда	Значение по умолчанию
A	Не используется	–
B	Метка блока, куда должен перейти транзакт	Ошибка

Пример: Транзакт должен перейти в блок с меткой Kol_1:

TRANSFER ,Kol_1

б). Статистический режим.

Формат блока:

TRANSFER A,[B],C

Операнд	Значение операнда	Значение по умолчанию
A	Вероятность передачи транзакта в блок C, задаваемая в долях тысячи	Ошибка
B	Метка блока, куда должен перейти транзакт с вероятностью 1–A	Следующий по порядку блок
C	Метка блока, в который должен перейти транзакт с вероятностью, указанной в операнде A	Ошибка

Пример. С частотой 0,275 транзакт переходит в блок с меткой Kol_1 и с частотой 0,725 – в блок с меткой Kol_2:

TRANSFER 0.725,Kol_1,Kol_2

в) Режим BOTH – Входящий транзакт пытается перейти к блоку, метка которого указана в операнде **B**. Если этот блок занят, то транзакт идет в блок с меткой, указанной в операнде **C**.

Пример. Автомобиль занимает любую свободную заправочную колонку **Z_1** или **Z_2**:

TRANSFER BOTH,Kol_1,Kol_2

.....

Kol_1 SEIZE Z_1

.....

Kol_2 SEIZE Z_2

.....

г). Режим ALL (все) – проверяется блок, метка которого указана в операнде **B**. Если этот блок занят, то последовательно проверяются все блоки до тех пор, пока не будет, достигнут блок, заданный в операнде **C**, или один из проверенных блоков не примет транзакт до достижения блока, метка которого указана в операнде **C**. Местоположение каждого последовательно проверяемого блока вычисляется путем добавления величины, указанной в операнде **D**. (если операнд **D** не используется, проверяется каждый блок между блоками, заданными в операндах **B** и **C**).

Формат блока:

TRANSFER ALL,B,C,D

где **B** – номер или метка блока;

C – номер или местоположение блока

D – приращение блока (по умолчанию = 1).

Пример. **TRANSFER ALL,Mesto_1,Mesto_N,3**

(в данном случае первым проверяется блок с меткой **Mesto_1**. Если это место занято, то проверяется каждый блок с местоположением на 3 позиции больше предыдущей. Если все блоки заняты, то проверка заканчивается на блоке с меткой **Mesto_N**. Если все блоки оказались занятыми, то транзакт снова начинает проверку с блока с меткой **Mesto_1** и т.д.)

Сохраняемые величины

В GPSS World разработчик модели может определить свои переменные, к которым можно обратиться из любого места модели, и начальные значения которых задаются перед началом моделирования. Такие переменные

называются **сохраняемыми величинами (ячейками)**. Совокупность логически связанных между собой ячеек образует **матрицу**. Информация, записанная в такую ячейку, сохраняется на протяжении всего периода моделирования. Чтобы задать сохраняемую величину, используется оператор **INITIAL** (инициализировать). Его стандартный числовой атрибут (СЧА) **Xj**, где *j* – имя ячейки. Этот оператор относится к командам GPSS World.

Формат оператора:

INITIAL A,B

Поле	Информация
Метка	не используется
Операция	INITIAL
Операнд А	Имя сохраняемой величины
Операнд В	Начальное значение величины

Пример.

INITIAL X\$OPLATA,5000 – ячейке OPLATA присваивается начальное значение 5000.

Если нужно обратиться, например, к ячейке OPLATA, куда ранее была занесена какая-то величина, то это делается так **X\$OPLATA**. Ячейка может иметь и числовое имя. В таком случае обратиться к сохраняемой величине, находящейся в ячейке с числовым именем 2, можно так: **X2**.

Если в модели требуются операции с матрицей, то вводится оператор описания матрицы – **MATRIX**.

Его формат:

<Метка> MATRIX A,B,C

Поле	Информация
Метка	Имя матрицы
Операция	MATRIX
Операнд А	Не используется
Операнд В	Число строк матрицы
Операнд С	Число столбцов матрицы

Пример. Матрица ZARPLATA состоит из двух строк и трех столбцов:
ZARPLATA MATRIX ,2,3

Если необходимо обратиться к элементу матрицы, то это делается с помощью СЧА **MX\$*j(m,n)*** , где *j* – имя матрицы, *m* – количество строк матрицы, *n* – количество столбцов.

Пример. Обратиться к элементу матрицы ZARPLATA, находящемуся в первой строке, втором столбце:

MX\$ZARPLATA(1,2)

Блок SAVEVALUE

При входе транзакта в блок **SAVEVALUE** (сохранить величину) значение сохраняемой величины можно изменить. Эта величина будет выведена в стандартном отчете о результатах моделирования.

Формат блока:

SAVEVALUE A [+,-],B

Операнд	Значение операнда	Значение по умолчанию
A	Номер или символьное имя сохраняемой величины	Ошибка
B	Величина, используемая для модификации (число или СЧА)	Ошибка

Примеры.

1. Увеличить сохраняемую величину Sebest на значение функции Materialy:

SAVEVALUE Sebest+,FN\$Materialy

2. В ячейке с именем Sebest сохранить длину очереди Ochered:

SAVEVALUE Sebest,Q\$Ochered

Проверка числовых выражений. Блок TEST

При помощи данного блока проводится сравнение величин, используя СЧА. Формат блока следующий:

TEST X A,B,[C]

Операнд	Значение операнда	Значение по умолчанию
X	Оператор отношения: G (больше) GE (больше или равно) E (равно)	A больше B ? A больше или равно B ? A равно B ?

	NE (не равно) LE (меньше или равно) L (меньше)	A не равно B? A меньше или равно B? A меньше B?
A	СЧА	Ошибка
B	СЧА	Ошибка
C	Имя блока, в который переходит транзакт при условии, условие не выполняется	Проверка выполняется в режиме отказа

Блок **TEST** может использоваться в режиме отказа или в режиме безусловного перехода.

Примеры.

1. Режим отказа.

TEST L Q\$Ocher_1,Q\$Ocher_2

(транзакт будет задержан в предыдущем блоке, пока очередь Ocher_1 не станет меньше очереди Ocher_2)

2. Режим условного перехода:

TEST G Q\$Ocher_1,Q\$Ocher_2,KOL

(транзакт перейдет в следующий после данного блок в случае, если длина очереди Ocher_1 будет больше длины очереди Ocher_2. В противном случае транзакт перейдет в блок с меткой KOL)

Переменные в GPSS

Переменные в GPSS бывают двух видов: арифметические и булевы. Арифметические переменные делятся на переменные с фиксированной точкой и плавающей точкой. Переменные с фиксированной точкой обозначаются оператором **VARIABLE**, а арифметические переменные с плавающей точкой – **FVARIABLE**. Формат операторов следующий:

<Метка> VARIABLE A

<Метка> FVARIABLE A

Операнд **A** отображает выражение, которое используется для вычисления значения переменной. Метка – это имя переменной (символьное или числовое). Для обращения к переменным, определяемым в модели ранее существует СЧА, обозначающий для переменной с фиксированной или плавающей точкой одинаково – **V**. Например, если необходимо обратиться к переменной *Cena*, которая была описана в модели ранее, то это делается путем использования СЧА следующим образом: **V\$Cena**. В выражении могут быть

записаны любые СЧА, например, СЧА очереди или функции и др. В выражениях используется следующие знаки:

- + сложение;
- вычитание;
- # умножение;
- / деление;
- @ деление по модулю;
- ^ возведение в степень;
- \ деление без остатка.

Примеры:

1. Вычислить значение переменной с фиксированной точкой:

Cena VARIABLE (300/7)#10

Ответ. Переменная с именем Cena равна 400 (40 умножить на 10).
Частное от деления 300/7 равно 40, так как дробная часть отбрасывается.

2. Вычислить значение переменной с плавающей точкой

Cena1 FVARIABLE (300/7)#10

Ответ. Переменная с именем Cena равна 428, так как частное от деления $300/700 = 42,85$. Затем $42,85 * 10 = 428,5$ и дробная часть отбрасывается.

3. **CenaOb FVARIABLE V\$Cena+V\$Cena1**

Эта запись означает, что переменная **CenaOb** равна сумме значений переменных **Cena** и **Cena1**, вычисляемых в программе ранее.

4. **PER FVARIABLE QT\$Ochered+5**

Переменная **PER** равна значению текущей длины очереди **Ochered** плюс пять.

Функции в GPSS World

Основными видами функций, рассматриваемых GPSS, являются дискретная числовая и непрерывная числовая. Реже используются табличная числовая, дискретная атрибутивная и табличная атрибутивная. Остановимся здесь на первых двух. Дискретная функция состоит из отдельных горизонтальных отрезков, непрерывная функция представляет собой кусочно-непрерывную функцию (Рисунок 3.4).

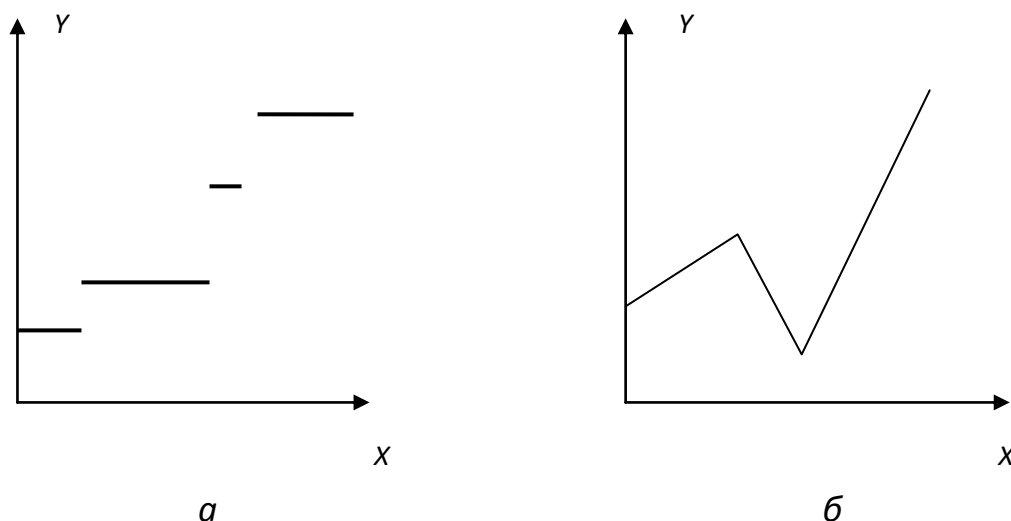


Рисунок. 9 – Дискретная (*a*) и кусочно-непрерывная (*б*) функции

Для обращения к функции из любого места модели и использования ее значения в различных выражения используется СЧА **FNj**, где *j* – имя функции.

Формат описания функций следующий:

<Метка> FUNCTION A,B

Операнд	Информация, задаваемая в поле
Метка	Имя функции (числовое или символьное)
Операция	FUNCTION
A	RNj (<i>j</i> =1,...,7) или СЧА
B	Dn или Cn (D определяет дискретную функцию, C – непрерывную; n для дискретных функций – количество горизонтальных отрезков, для непрерывной – это число, на единицу больше числа отрезков, составляющих функцию (число точек))

В операторе описания функций при описании координат значения *X* должны располагаться по возрастанию. Координаты пограничных точек записываются следующей строки после FUNCTION.

Примеры:

1). Смоделировать дискретную случайную переменную, заданную в следующей таблице:

Значение случайной переменной (Y)	Относительная частота (X)	Суммарная частота
3	0,15	0,15
5	0,60	0,75
2	0,25	1,00

Ответ: **PRIB FUNCTION RN3,D4**

.15,3/.6,5/1,2

Графически это будет выглядеть следующим образом (Рисунок 10):

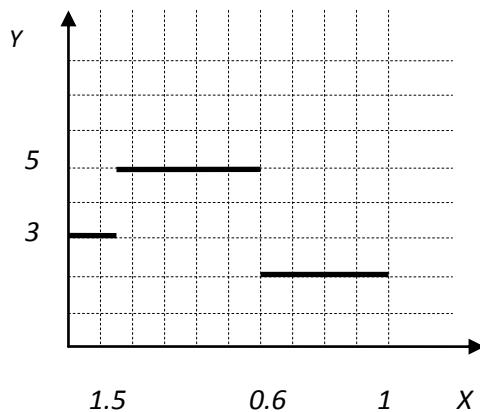


Рисунок 10 – Графическое изображение дискретной функции

2). Смоделировать случайную функцию, равномерно распределенную на интервале [3,7] (Рисунок 11) .

CEB_T FUNCTION RN2,C2

0,3/1,7

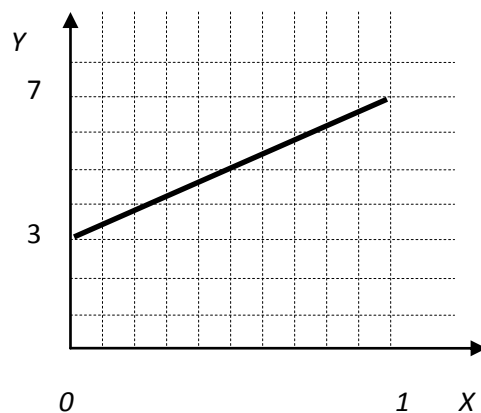


Рисунок 11 – Графическое изображение непрерывной функции

3). Переменная Skor_1 равна значению функции Sebest , умноженному на значение функции Koli.

Skor_1 FVARIABLE N\$Sebest # FN\$Koli

В GPSS World для моделирования некоторых вероятностных функций можно использовать инструмент встроенных функций (всего 24 вероятностных функций). Это такие как распределения как распределение Пуассона (обращение к ней через обозначение **Exponential**), нормальное распределение, гамма-распределение, распределение Вейбулла, биномиальное и др.

Пример. Транзакты поступают согласно экспоненциальному закону распределения вероятностей (то есть распределению Пуассона) с параметрами $\lambda=0$ и $\beta=35$:

GENERATE (Exponential (1,0,35)).

2.4. Имитационный проект

2.4.1. Формулирование и описание проблемы

(постановка задачи)

Пример¹.

Структурным подразделением автотранспортного предприятия является автозаправочная станция, работающая на условиях хозрасчета. На АЗС имеется две заправочные колонки и известны следующие параметры работы АЗС:

поток автомобилей, поступающих на заправку, приходит с интенсивностью 5 ± 2 минуты;

время заправки на первой колонке составляет $10 \pm 2,5$ мин, а на второй – 13 ± 4 мин;

автомобиль подъезжает к свободной колонке.

Требуется смоделировать работу АЗС в течение рабочей смены (8 ч) и определить параметры функционирования АЗС:

коэффициент загрузки каждой колонки;

среднее время обслуживания в каждой колонке;

максимальное, среднее и текущее число автомобилей в очереди к каждой колонке;

среднее время нахождения автомобиля в каждой очереди и др.;

прибыль от работы АЗС за день, если прибыль от заправки одной машины составляет 80 р.

У предприятия существует возможность введения более производительной техники на автозаправочной станции. Требуется статистически оценить целесообразность внедрения более производительной техники (существенность сокращения очереди), а именно, замену первой колонки более

¹ Здесь и далее курсивом напечатан пример. Комментарии к нему даны стандартным шрифтом. Пример от комментариев отделяется звездочками.

совершенной, производящей заправку за 8 мин².

2.4.2. Создание концептуальной модели

Концептуальная модель – это описание в словесной или графической форме элементов оригинала (моделируемой системы), связей между ними (элементами), процессов протекающих в оригинале, влияние внешней среды.

При разработке концептуальной модели производится статическое и динамическое описание системы. Определяют границы системы, приводятся описание внешней среды, выделяют существенные элементы и дают их описание, формируют переменные, параметры, функциональные зависимости как для отдельных элементов и процессов, так и для всей системы, ограничения, целевые функции (критерии). На данном этапе выдвигаются гипотезы и фиксируют все допущения (предположения), необходимые для построения имитационной модели. Обсуждают уровень детализации моделируемых процессов.

В концептуальное описание системы должны быть включены критерии эффективности функционирования системы и оцениваемые альтернативные решения, которые рассматриваются как входы модели. Здесь же уточняются основные переменные модели, участвующие в ее описании.

При создании концептуальной модели производят исключение второстепенных элементов, группировку оставшихся элементов в блоки по функциональному признаку, окончательно принимают предположения и гипотезы о законах распределения переменных (если они заданы).

Основная проблема при этом заключается в нахождении компромисса между простотой модели и ее адекватностью исследуемой системе. Четко установленных правил концептуального описания модели не существует. Основным требованием на этом этапе является четкое и ясное отражение моделируемой системы. Поэтому данный раздел проекта выполняется в произвольной форме. Если моделируемый процесс представляет собой систему массового обслуживания (СМО), то концептуальная модель должна быть описана графически в общепринятых терминах и обозначениях СМО.

Данный пункт включает следующие подпункты:

3.1. Подробная постановка задачи (обоснование целесообразности построения модели, формулировка цели моделирования, описание реальной си-

² Согласно закону больших чисел при большом количестве испытаний будет доказана существенность изменения результативного показателя в результате данного нововведения. Однако для наглядности примера и в целях упрощения оценим статистически именно данный показатель.

стемы или проблемной ситуации);

3.2. Содержательное описание объекта моделирования (перечисление основных гипотез, выдвинутых при построении модели).

3.3. Функциональная схема объекта моделирования.

3.4. Список параметров и переменных модели.

3.5. Перечень искомых выходных данных модели (статистика).

3.6. Функциональные зависимости, используемые в концептуальном описании.

Процедура заправки автомобилей является одной из составляющих единого производственного процесса на автотранспортном предприятии. Поэтому оптимизация процесса заправки автомобилей позволит сократить до минимума время пребывания автомобиля на АЗС и повысит эффективность его работы. Оптимизация обслуживания на АЗС выражается в сокращении времени простоя в очереди на заправку. Так как процесс заправки является динамическим и производится параллельно функционирующими устройствами, необходимо для моделирования производственного процесса использовать метод имитационного моделирования.

Особенности функционирования АЗС состоят в том, что ее деятельность полностью зависит от входящего потока автомобилей. Таким образом, чем больше автомобилей будет поступать на заправку, тем больше будут очереди на обслуживание, и выше уровень загрузки колонок. Поэтому очень многое зависит от закона поступления автомобилей на заправку.

Очень важным моментом является общее время пребывания автомобиля на АЗС, которое зависит от длины очереди и времени обслуживания на конкретной колонке. Колонки имеют неодинаковую производительность работы, и, следовательно, будут иметь разный коэффициент загрузки, определение которого тем более важно, что поставлен вопрос о замене одной из колонок на более производительную.

Целью моделирования работы АЗС является создание адекватной модели динамического процесса и проведение экспериментов с моделью для оптимизации процесса обслуживания автомобилей, что увеличит экономическую эффективность функционирования предприятия.

Представим процесс функционирования в общем виде (Рисунок 12 и 13):

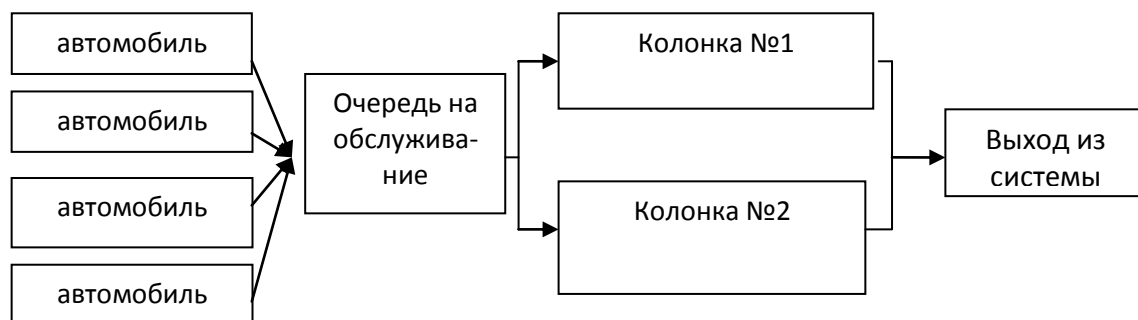


Рисунок 12 – Общая схема работы АЗС

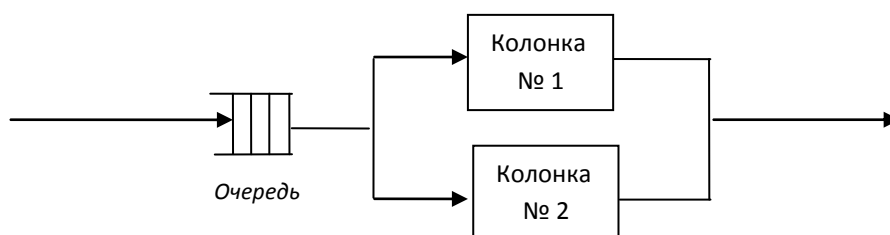


Рисунок 13 – Структурная схема процесса заправки автомобилей

Автомобили подъезжают к автозаправочной станции каждые 5 ± 2 мин.

Дисциплина обслуживания – «первый поступил – первый обслужен».

Время обслуживания в колонке №1 составляет $10 \pm 2,5$ мин, а в колонке № 2 – 13 ± 4 мин.

Критерием эффективности в системе является минимум времени на обслуживание.

Сокращение времени пребывания автомобиля на АЗС возможно за счет сокращения очереди вследствие введения дополнительных колонок или их замены на более производительные. Это связано с дополнительными денежными расходами, в то же время, сокращение времени пребывания в очереди за счет нововведений может быть несущественным. Поэтому необходимо статистически оценить значимость нововведений. Это делается с помощью применения известного алгоритма статистического анализа ANOVA, процедура которого является встроенной в GPSS World. Процедура требует выдвижения первоначальной (нулевой) гипотезы, которая может быть отвергнута или принята в результате анализа.

Гипотеза. Если ввести автоматическое обслуживание на одной из колонок, что приведет к сокращению времени обслуживания с 10 до 8 минут, существенно ли это уменьшит длину очереди? Стоит ли принимать такое управленческое решение?

2.4.3. Формализация имитационной модели

На этом этапе модель, представленная как система массового обслуживания, воплощается в конкретную машинную модель, ориентированную на использование конкретных программно–технических средств.

Язык моделирования GPSS имеет блочную концепцию структуризации, структура моделируемого процесса изображается в виде потока транзактов, проходящего через обслуживающие устройства, очереди и другие элементы систем массового обслуживания.

Удобной формой представления логической структуры машинной модели является схема алгоритма. Она отражает логику функционирования моделируемого объекта и содержит последовательность блоков. Общие требования к изображению схем алгоритмов приведены в ГОСТ 19.003 – 80 «Единая система программной документации» (Рисунок 14). В системе GPSS World приняты свои специальные обозначения.

Для представления логической структуры машинной модели приведем схему алгоритма. Она отражает логику функционирования моделируемого объекта и содержит последовательность блоков.

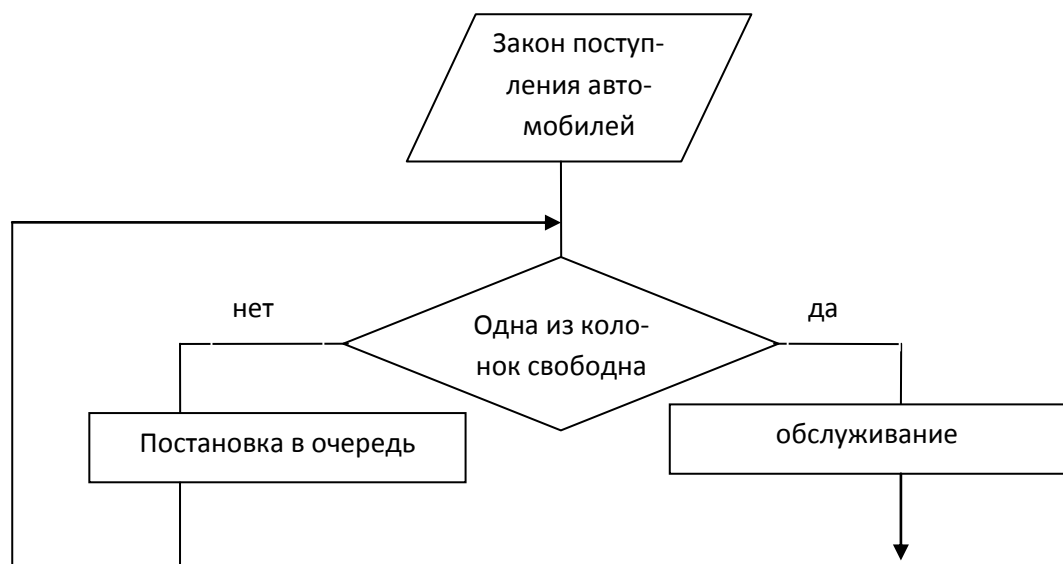


Рисунок 14 – Блок-схема обслуживания на АЗС в соответствии с ГОСТ

Кроме того, в некоторых средствах имитационного моделирования имеется возможность построения логической структуры реальной экономической системы с помощью соответствующих алгоритмических изображений. Так, в GPSS World блок-схему функционирования АЗС можно представить в следующем виде (Рисунок 3.10):

2.4.4. Подготовка исходной информации

После постановки задачи необходимо сформулировать требования к исходной информации об объекте моделирования. Затем принимают предположения и гипотезы о достаточности информации для решения задачи, об ограничениях на ресурсы при решении задачи, об ожидаемых результатах моделирования.

Кроме этого необходимо определить параметры системы, входные и выходные переменные, факторы воздействия внешней среды.

Описание каждого параметра и переменной дается в следующем порядке:

- 1) определение и краткая характеристика;
- 2) символ обозначения и единица измерения;
- 3) диапазон изменения (для переменных);
- 4) место применения в модели.

Представленная система относится к системам массового обслуживания с 2-мя обслуживающими устройствами. Это – разомкнутая система массового обслуживания.

Динамическими элементами системы являются – автомобили.

Обслуживающие устройства – колонки АЗС (табл. 2).

Таблица 2 – Определение основных входных данных

Элемент системы	Роль в системе	Обозначение
Автомобиль	Транзакт	—
Колонки: №1 №2	Устройства обслуживания	KOLONKA_1, KOLONKA_2
Очередь	—	ZAPRAVKA

2.4.5. Программирование модели

На данном этапе концептуальное или формальное описание модели сложной системы преобразуется в программу–имитатор в соответствии методикой программирования GPSS (или с помощью других программных средств).

Если схема алгоритма составлена подробно, то написание программы не составит большого труда. Этот шаг заключается в записи пространственной структуры в линейном виде. Если необходимо, в программу включаются комментарии, описывающие объекты моделирования: функции, переменные, ячейки и т.п.

После того, как программа написана, ее нужно отладить и проверить с помощью тестовых задач. В окне Journal выдаются сообщения об ошибках.

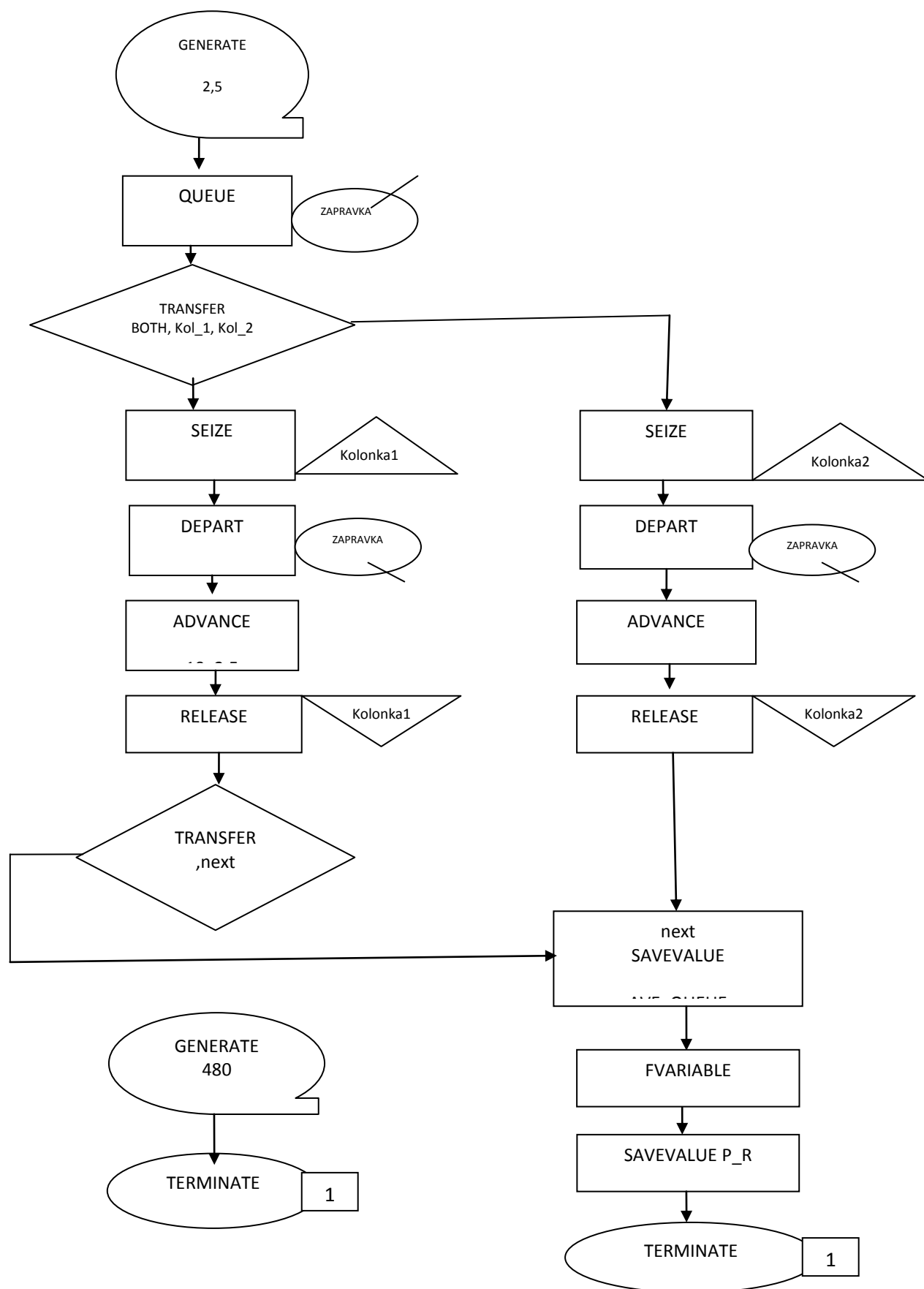


Рисунок 15 – Блок-схема обслуживания на АЗС
в соответствии с принятыми обозначениями в GPSS

Программа на языке GPSS, описывающая процесс работы АЗС:

```
GENERATE 5,2 ; равномерное поступление автомобилей на заправку
QUEUE ZAPRAVKA ; автомобиль занимает очередь
TRANSFER BOTH,KOL_1,KOL_2 ; переход к одной из двух колонок
KOL_1 SEIZE KOLONKA_1 ; занимает колонку№1
DEPART ZAPRAVKA ; выходит из очереди
ADVANCE 10,2.5 ; процесс заправки занимает 10±2.5мин.
RELEASE KOLONKA_1 ; автомобиль освобождает колонку
TRANSFER ,NEXT ; переход к оператору с меткой NEXT
KOL_2 SEIZE KOLONKA_2 ; занимает колонку№2
DEPART ZAPRAVKA ; выходит из очереди
ADVANCE 13,4 ; процесс заправки занимает 13±4мин
RELEASE KOLONKA_2 ; автомобиль освобождает колонку
NEXT SAVEVALUE AVE_QUEUE,QT$ZAPRAVKA
PRIBY FVARIABLE N$NEXT#80 ; общая прибыль
SAVEVALUE P_R,V$PRIBY ; сохраняемая в отчете величина PRIBY
TERMINATE ; удаление транзакта из модели

GENERATE 480 ; время моделирования работы АЗС равно 8 часов
TERMINATE 1
START 1
```

2.4.6. Анализ и интерпретация результатов пробного прогона модели

Правильная интерпретация результатов не менее важна, чем анализ исходных данных, так как от этого зависят выводы исследователя по функционированию моделируемой системы. Сначала выделяют те результаты, которые нужны для дальнейшего анализа. Затем эти результаты должны быть интерпретированы относительно моделируемого объекта, т.е. должен быть осуществлен переход от информации, полученной в результате машинного эксперимента с моделью, к информации применительно к объекту моделирования.

По полученным результатам должна быть проведена проверка гипотезы и предположений, сделаны соответствующие выводы и даны рекомендации по практическому использованию результатов моделирования. На данном этапе приводятся результаты компьютерных экспериментов в виде графиков, таблиц, распечаток.

По результатам моделирования получен стандартный отчет, который выдает основные характеристики системы.

Проанализируем стандартный отчет и определим основные характеристики функционирования системы. В отчете указывается, что модельное

время (*END TIME*) составляет 480 ед., в данном случае – минут. Число каналов обслуживания (*FACILITY*) равно двум (две колонки).

Первая колонка за день обслуживает 48 автомобилей (*ENTRIES* = 48), вторая – 37 автомобилей, так как время обслуживания на второй колонке больше.

Важной характеристикой является коэффициент загрузки устройств обслуживания (коэффициент использования узла (*UTIL*)), который рассчитывается как соотношение общего времени обслуживания данным устройством к общему времени наблюдения. Значение коэффициента, близкое к единице свидетельствует о высокой загрузке устройства в течение времени моделирования (в данной модели этот коэффициент для первой колонки составляет 0,984, для второй – 0,974).

Среднее время обслуживания (*AVE.TIME*) первой колонкой равно 9.835 мин, второй – 12.636 мин.

К моменту окончания моделирования оба устройства были доступны (*AVAIL.=1*). Так как автомобили обслуживали в порядке общей очереди, и условия приоритетности не было установлено, то количество автомобилей, ожидающих выполнения операции с прерыванием обслуживания других автомобилей (*PEND*, равно 0. Соответственно, количество автомобилей, обслуживание которых было прервано (*INTER*), также равно 0. Количество автомобилей, ожидающих выполнения специфического условия, зависящего от состояния колонок (*RETRY*) равно 7 и 7. Отказов в обслуживании каких-либо автомобилей (*DELAY=0*) не было.

Данные стандартного отчета о состоянии очереди на заправку можно проанализировать следующим образом:

Максимальное содержание очереди (*MAX*) равно 8 автомобилей. Текущее содержание очереди к моменту завершения процесса моделирования (*CONT.*) равно 7. Общее количество автомобилей, входящих в очередь (*ENTRY*), равно 92. Общее число автомобилей, входящих в очередь с нулевым временем пребывания в ней (*ENTRY(0)*), равно 4. Взвешенная по времени средняя длина очереди в течение периода моделирования (*AVE.CONT.*) равно 3,392 мин. Среднее время нахождения автомобиля в очереди (*AVE.TIME*) составляет 17,639 мин.

Таблица 3 – Стандартный отчет результатов моделирования

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	17	2	0

NAME	VALUE
AVE_QUEUE	10004.000
KOLONKA_1	10002.000
KOLONKA_2	10003.000
KOL_1	4.000
KOL_2	9.000
NEXT	13.000
PRIBY	10000.000
P_R	10005.000
ZAPRAVKA	10001.000

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	92	0	0			
	2	QUEUE	92	0	0			
	3	TRANSFER	92	7	0			
KOL_1	4	SEIZE	48	0	0			
	5	DEPART	48	0	0			
	6	ADVANCE	48	1	0			
	7	RELEASE	47	0	0			
	8	TRANSFER	47	0	0			
KOL_2	9	SEIZE	37	0	0			
	10	DEPART	37	0	0			
	11	ADVANCE	37	1	0			
	12	RELEASE	36	0	0			
NEXT	13	SAVEVALUE	83	0	0			
	14	SAVEVALUE	83	0	0			
	15	TERMINATE	83	0	0			
	16	GENERATE	1	0	0			
	17	TERMINATE	1	0	0			

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KOLONKA_1	48	0.984	9.835	1	86	0	0	7	0
KOLONKA_2	37	0.974	12.636	1	85	0	0	7	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
ZAPRAVKA	8	7	92	4	3.392	17.698	18.503	0

SAVEVALUE	RETRY	VALUE
AVE_QUEUE	0	17.639
P_R	0	6640.000

Среднее время пребывания в очереди одного автомобиля в течение периода моделирования без учета «нулевых» входов (AVE.(-0)) составило

18,503 мин. И количество автомобилей, ожидающих выполнения специфического условия, зависящего от состояния очереди (RETRY), составило 0.

Значение сохраняемой величины AVE_QUEUE (время пребывания в очереди на заправку) равно 17,698 мин.

Как показал анализ результатов моделирования, значительную часть времени автомобили находятся в очереди на заправку. Управленческое решение, которое может быть принято в данном случае – это увеличение количества обслуживающих устройств (колонок) или замена колонок на работающие более быстро. По условиям задачи руководитель АЗС имеет возможность, осуществив дополнительные капиталовложения, заменить колонку, обслуживающую в среднем за 10 мин, на более совершенную, производящую эту же операцию в среднем за 8 мин. Выясним, как это повлияет на выходные характеристики модели, заменив в операторе ADVANCE 10,2.5. время операции (ADVANCE 8,2.5). После осуществления прогона модели получим следующий отчет (табл. 4).

Таблица 4 – Стандартный отчет о результатах моделирования

GPSS World Simulation Report – Untitled Model 1.3.1									
Thursday, March 03, 2005 18:09:07									
START TIME	END TIME		BLOCKS	FACILITIES		STORAGES			
0.000	480.000	16	2	0					
FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
KOLONKA_1	57	0.956	8.053	1	95	0	0	0	0
KOLONKA_2	37	0.965	12.514	1	93	0	0	0	0
QUEUE	MAX CONT.		ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(–0)		RETRY
ZAPRAVKA	3	0	94	11	0.909	4.644	5.259	0	
SAVEVALUE	RETRY		VALUE						
AVE_QUEUE	0		4.644						

Оформим результаты моделирования в следующую таблицу (табл.5), выделив основные характеристики.

Таблица 5 –Результаты моделирования работы АЗС в течение смены

Параметры	Среднее время заправки на первой колонке, мин	
	10	8
Количество обслуженных автомобилей, всего	85	94
В т. ч.		
колонкой №1	48	57
колонкой №2	37	37
Среднее время обслуживания:		
колонка №1	9.835	8.053
колонка №2	12.636	12.514
Коэффициент загрузки:		
колонка №1	0.984	0.965
колонка №2	0.974	0.965
Максимальное число автомобилей в очереди	8	3
Среднее число автомобилей в очереди	3.392	5
Среднее время ожидания заправки, мин	17.698	4.644

Дополнительно можно проанализировать работу системы с использованием графика и гистограммы.

Для получения графика изменения какого-либо показателя работы системы (например, количество автомобилей в очереди на заправку) заполняются поля окна *Edit Plot Window* (Рисунок 16).

Рисунок 16 – Ввод исходных данных для построения графика

График имеет следующий вид (Рисунок 17).

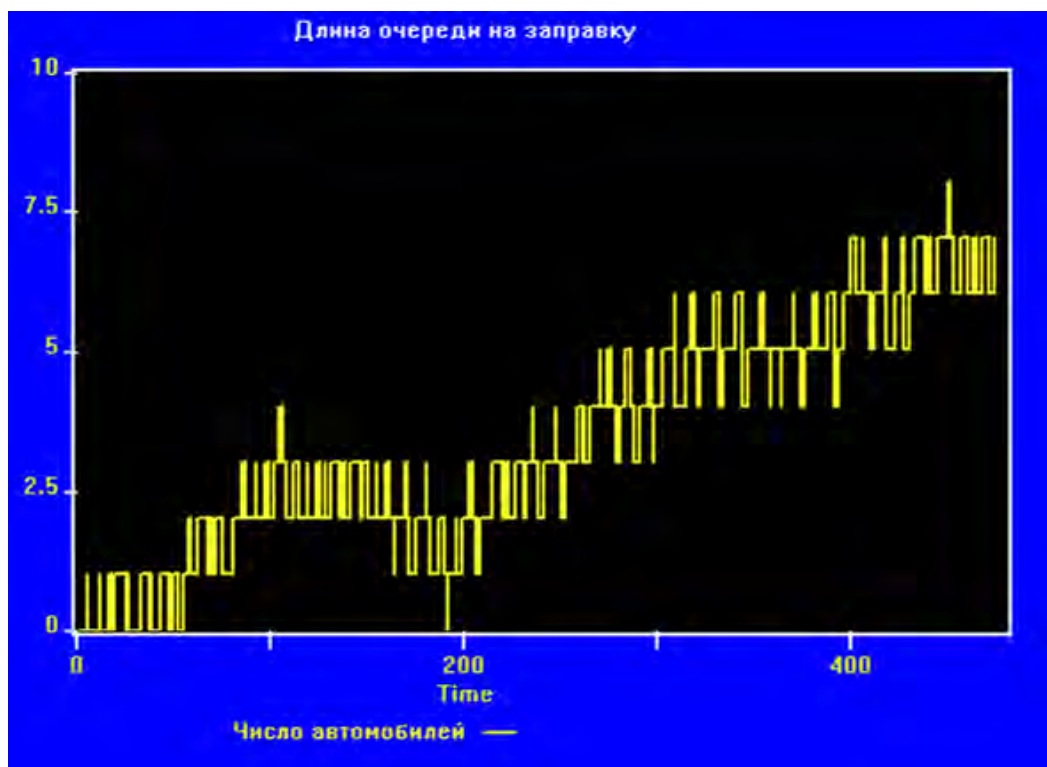


Рисунок 17 – Графическое изображение результатов моделирования

График позволяет установить, что в период между 360...370 минутами с начала работы автозаправочной станции в очереди скапливается наибольшее число автомобилей – десять.

Дополнительно может быть построена гистограмма результатов моделирования (Рисунок 18). Это делается путем добавления к тексту основной программы команды *QTABLE*:

INFORM QTABLE ZAPRAVKA,0,3,35

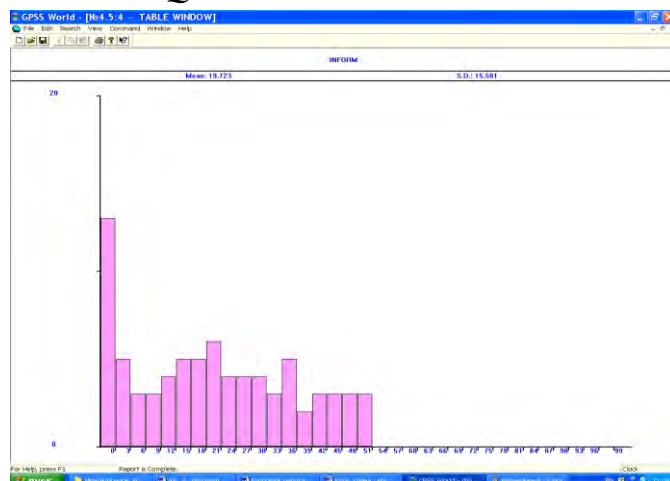


Рисунок 18 – Гистограмма результатов моделирования

2.4.7. Исследование свойств имитационной модели

Полученные с помощью имитационной модели результаты должны обладать требуемой точностью и достоверностью. Достоверность результатов моделирования предполагает, что имитационная модель отвечает некоторым специфическим требованиям, позволяющим оценить качество модели. Оценка качества модели предполагает ее проверку на соответствие целям моделирования.

Оценка свойств модели включает:

оценку адекватности модели;

оценку точности результатов моделирования (погрешности имитации, обусловленную наличием в модели неидеальных генераторов случайных чисел);

определение длительности переходного режима в работе имитационной модели;

оценку устойчивости результатов имитации исследуемых процессов;

исследование чувствительности модели.

2.4.7.1. Оценка адекватности имитационной модели

и точности результатов моделирования

Оценка *адекватности* модели отражает степень ее соответствия реальной системе. Она может быть проведена **только**, если у исследователя есть возможность сопоставить результаты прогона модели с работой реальной системы. Алгоритм проверки адекватности модели состоит в сравнении выходов модели и реальной системы при одинаковых значениях входов. При этом как значения откликов системы, так и значения откликов модели считаются *выборочными данными*. В этом случае используются статистические методы проверки гипотез. Нулевая гипотеза (H_0) предполагает несущественность различий между средними величинами откликов системы и откликов модели. При этом могут использоваться различные статистические критерии, например, критерий t -Стьюдента. В результате проверки гипотезы она может быть отвергнута, тогда принимается альтернативная гипотеза.

Если же проводится моделирование проектируемого объекта, то оценка адекватности проведена быть не может.

Оценка точности результатов моделирования

Входные параметры модели часто задаются вероятностно (в виде определенного интервала). Генератор псевдослучайных чисел, встроенный в известные системы имитационного моделирования, определяет какое-либо значение параметра (например, поступление транзактов) внутри этого интервала. Наличие генератора является источником погрешности имитации.

Поэтому проводят несколько прогонов модели (обычно $N = 10$) при одних и тех же **серединных** значениях параметров X , но разных пограничных точках интервалов (например, интервала поступления транзактов в блоке GENERATE). По полученным откликам (Y) оценивается погрешность. Оценить погрешность можно путем определения оценок математического ожидания и дисперсии отклонения компонент вектора откликов Y (применяется понятие «оценка математического ожидания», а не «математическое ожидание», так как речь идет о выборочной совокупности, приблизительно, оценочно отражающей генеральную совокупность). Оценка математического ожидания (\bar{Y}) вычисляется по формуле:

$$\bar{Y} = \frac{1}{N} \sum_{k=1}^N Y_k ,$$

где Y_n – отклики модели по прогонам, N – число прогонов.

Оценка дисперсии вычисляется по формуле:

$$D_n = \frac{1}{N-1} \sum_{k=1}^N (Y_k - \bar{Y}_n)^2 .$$

Далее определяется доверительный интервал математического ожидания отклика Y .

Так как погрешности порождаются генераторами псевдослучайных чисел, можно допустить нормальный закон распределения отклонений Y_k от значений Y . Так как выборка является малой ($k < 30$), то для нахождения доверительных интервалов используется t -статистика (распределение Стьюдента). Алгоритм нахождения доверительного интервала (интервала, в котором с выбранной исследователем вероятностью находится математическое ожидание значений отклика) следующий:

$$\bar{Y}_n - \varepsilon \leq Y \leq \bar{Y}_n + \varepsilon , \quad (\varepsilon = t_{0.05} \sqrt{D_n / (N-1)})$$

где $t_{0,05}$ – табличное значение, определяемое по таблице «Значения коэффициента t -Стьюдента» (оно зависит от числа степеней свободы ($\nu = N-1$) и уровня значимости суждений $\alpha = 0,05$).

2.4.7.2. Определение объема выборки (числа прогонов)

Выше было сказано о необходимости проведения нескольких прогонов имитационной модели. Возникает вопрос – сколько прогонов нужно провести, чтобы сформировать выборочную совокупность значений определенного параметра, по которой можно найти доверительный интервал?

Точность суждений о значении какого-либо показателя отклика (например, средней длины очереди) зависит от количества выборочных значений: чем выборка больше, тем точнее результаты. Если точность (т.е. доверительный интервал) задается заранее и исследователь остановился на каком-то одном уровне вероятности суждения (чаще P берется равным 0,95), необходимое количество прогонов (N) определяется путем обратной задачи следующим образом:

$$N \cong N_0 \frac{d_0^2}{d^2}, \text{ где}$$

N_0 – число пробных прогонов;

d_0 – длина получившегося по результатам пробных прогонов доверительного интервала (в единицах измерения оцениваемого показателя);

d – экспертно определяемая необходимая для исследователя длина доверительного интервала (в единицах измерения оцениваемого показателя).

Проведем десять прогонов ($N = 10$) модели, изменяя границы интервалов поступления автомобилей, но при неизменном срединном значении интервала, равного 5 мин:

Значение операнда B блока GENERATE	1,5	1,7	1,8	1,9	2,0	2,5	2,7	2,8	2,9	3,0
Отклик (результат прогона – средняя длина очереди на заправку, ед.)	3,354	2,555	3,077	3,637	3,392	6,429	6,448	2,330	2,254	4,247

По результатам прогонов:

$$1) \bar{Y} = \frac{1}{N} \sum_{k=1}^N Y_k = 3,772$$

$$2) D_n = \frac{1}{N-1} \sum_{k=1}^N (Y_k - \bar{Y}_n)^2 = \frac{21,134}{9} = 2,347$$

$$3) t_{0,05} \text{ по таблице } t\text{-Стьюдента при } (v = 9) = 2,15$$

4) доверительный интервал

$$3,372 - t_{0,05} \sqrt{D_n / (N-1)} \leq Y \leq 3,372 + t_{0,05} \sqrt{D_n / (N-1)}$$

$$\text{или } 3,372 - 2,15 \cdot \sqrt{2,347/9} \leq Y \leq 3,372 + 2,15 \cdot \sqrt{2,347/9}.$$

Таким образом, по результатам пробных прогонов средняя длина очереди с вероятностью 0,95 находится в интервале от 2,86 до 3,88.

Примем, что это для нас недостаточно конкретное значение и сузим интервал до следующих значений: [3,06; 3,68]. Тогда согласно формуле

$$N \cong N_0 \frac{d_0^2}{d^2} \text{ находим } N \cong 10 \frac{1,04}{0,38} = 27 \text{ прогонов.}$$

2.4.7.3. Оценка устойчивости модели

Под *устойчивостью* результатов имитации понимается степень нечувствительности ее к изменению входных условий. Устойчивость модели – это ее способность сохранять адекватность на всем диапазоне рабочей нагрузки, а также при внесении изменений в конфигурацию системы. Например, при увеличении работы системы с пяти до восьми часов будет ли разработанная модель с достаточной точностью отражать работу системы? Чем ближе структура модели соответствует структуре системы и выше степень ее детализации, тем выше устойчивость модели.

В целом устойчивость результатов моделирования можно оценить *дисперсией значений отклика* (одного из показателей работы системы, например, коэффициента загрузки устройства). Если при увеличении времени моделирования дисперсия отклика не увеличивается, то результаты работы данной модели устойчивы.

Для получения первой выборочной статистической совокупности, устанавливается какое-либо модельное время, например, пять часов. Затем выбирается некий шаг для контроля величины параметра работы системы, допустим, каждые 30 мин. Выборочная совокупность будет состоять из десяти

ти значений. Проводится расчет дисперсии (D_1). Затем модельное время увеличивается, например, с пяти до восьми часов, и снова осуществляется прогон модели. Новая выборочная совокупность содержит уже 16 значений. Рассчитывается новая дисперсия (D_2). Затем все рассчитанные дисперсии сравниваются между собой.

Чтобы результаты моделирования были устойчивыми, дисперсии должны различаться несущественно. Рост разброса контролируемого параметра от начального значения при изменении числа шагов указывает на неустойчивый характер имитации исследуемого процесса. Четко установленной методики для процедуры проверки устойчивости модели не существует.

Можно установить границы колебания контролируемого параметра экспертным путем и если данный параметр выходит за пределы колебаний, то на данном временном интервале констатируется потеря устойчивости результатов моделирования.

Реже для проверки существенности различия дисперсий используется критерий Бартлетта, расчетное значение которого определяется по формуле:

$$B = \frac{1}{c} \left(\int \ln S^2 - \sum_{i=1}^k f_i \ln S_i^2 \right), \text{ где}$$

$$S^2 = \frac{1}{f} \sum_{i=1}^k f_i S_i^2$$

$$f = \sum_{i=1}^k f_i, \quad f_i = n - 1$$

Методика проверки статистической гипотезы следующая:

1) Выдвигается нулевая гипотеза H_0 о *несущественности расхождений дисперсий значений откликов*.

2) По вышеуказанной формуле рассчитывается фактическое значение критерия Бартлетта ($B_{\text{факт}}$).

3) Определяется теоретическое (табличное) значение критерия Бартлетта ($B_{\text{табл}}$).

4) Сравниваются $B_{\text{факт}}$ и $B_{\text{табл}}$. Если фактическое значение превышает табличное, то гипотеза о несущественности расхождений дисперсий значений откликов отвергается и принимается противоположная гипотеза (т.е. характер имитации неустойчив), если наоборот – нулевая гипотеза принимается.

Для оценки устойчивости модели может быть также использован статистический критерий Вилкоксона.

Выберем шаг, по которому будем формировать статистическую совокупность значений длины очереди на заправку, равным 30 мин. Проведем прогон модели со временем 3 часа (180 мин) (Рисунок 19).

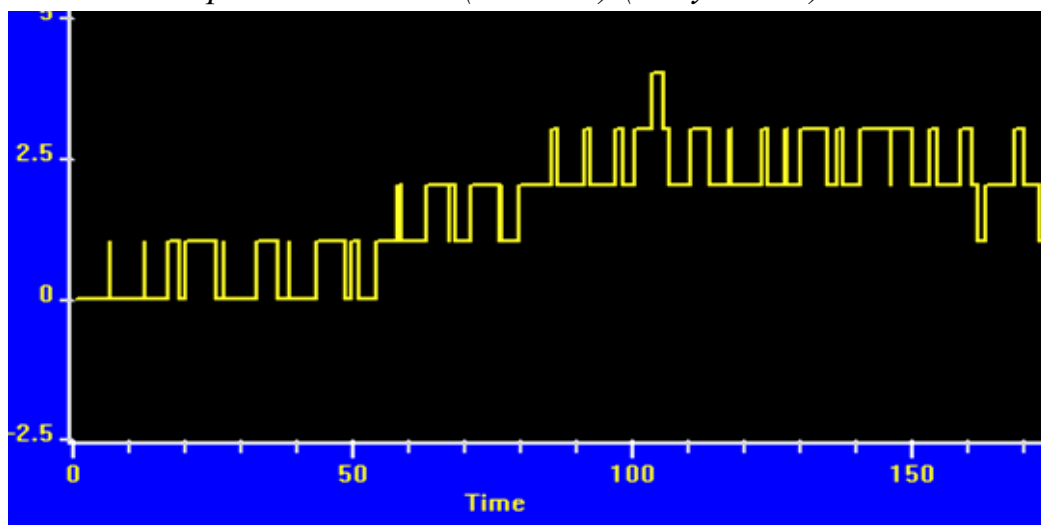


Рисунок 19 – Графические результаты прогона модели (модельное время 180 мин)

Зафиксируем длину очереди через каждые 30 мин.: 0; 0,5; 1,2; 1,2; 2,5; 1,2. Дисперсия по этим точкам равна $D_n = \frac{1}{N-1} \sum_{k=1}^N (Y_k - \bar{Y}_n)^2 = 0,71$.

Далее проведем прогон модели с модельным временем 300 мин (Рисунок 20).

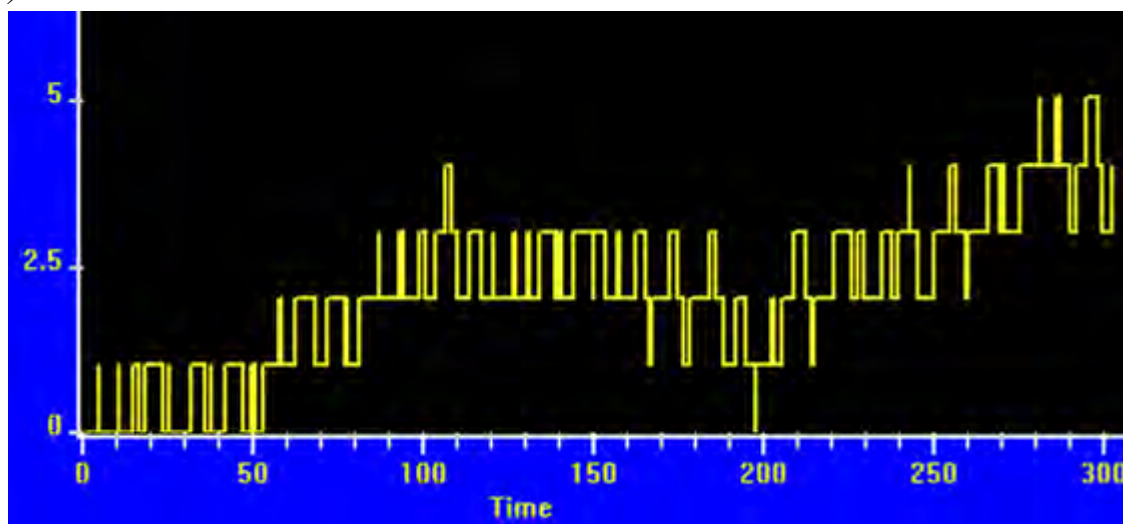


Рисунок 20 – Графические результаты прогона модели (модельное время 300 мин)

Зафиксируем длину очереди через каждые 30 мин.: 0; 0,5; 1,2; 1,2; 2,5; 1,2; 3; 4; 2,3; 5.

Дисперсия по этим точкам равна $D_n = \frac{1}{N-1} \sum_{k=1}^N (Y_k - \bar{Y}_n)^2 = 2,49$.

Проведя экспертную оценку существенности различий дисперсий, мы видим, что они сильно (в 3,5 раза) различаются между собой. Поэтому можно констатировать, что при увеличении периода моделирования с трех до пяти часов модель теряет устойчивость.

2.4.7.4. Определение длительности переходного режима

Имитационные модели создаются для изучения системы в типичных для нее условиях. В условиях реальной действительности требуется некоторое время T_0 для достижения моделью необходимого установившегося состояния, т.е. начальный период моделирования («время разогрева») не является типичным для работы системы. Поэтому исследователь должен принять меры по уменьшению влияния этого периода времени на результаты моделирования или вообще по исключению его из результатов моделирования.

При этом можно воспользоваться одним из трех способов: использование длинных прогонов модели; исключение из рассмотрения начального периода прогона; выбор такого начального условия, которое ближе всего к типичному, и, тем самым уменьшает длительность переходного режима. Первый способ применяется, если прогон модели не требует много машинного времени. Недостатком второго способа является то, что не всегда легко установить, когда заканчивается переходный режим. При использовании третьего способа проводят пробный прогон, чтобы определить момент выхода системы в стационарный режим.

Под *стационарным режимом системы* понимается такое состояние, в котором противодействующие влияния сбалансированы и компенсируют друг друга. То есть модель находится в равновесии, когда отклик (например, длина очереди или объем продаж) не выходит за некие предельные значения. Результаты прогона модели дают, как правило, несколько возможных откликов (результативных показателей). Так как эти показатели могут войти в стационарный режим не одновременно, требуется наблюдение за каждым из них. Стационарный режим работы для всей системы устанавливается по тому из откликов, который последним входит в режим стационарного состояния. На Рисунке 21 представлен визуальный способ определения начала стационарного режима.

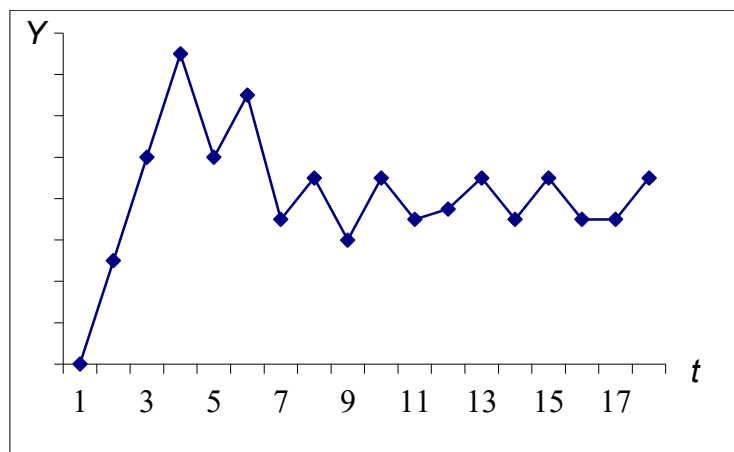


Рисунок 21 – *Пример изменения во времени сезонных объемов продаж предприятия (стационарный режим начинается через семь единиц модельного времени после начала работы)*

Существуют статистические способы определения стационарности состояния системы. Один из них – с помощью критерия Вилкоксона. В этом случае сравниваются соседние значения средних значений отклика l_k и l'_k ($k = \overline{1, m}$), где m – объем выборки, k – номера замеров значений отклика. Далее выдвигается гипотеза H_0 об однородности этих совокупностей, которая проверяется по критерию Вилкоксона. Если фактическое значение критерия находится в пределах критического, определяемого на основе специально рассчитанных таблиц, то гипотеза об однородности выборок подтверждается, т.е. на данном временном интервале система находится в режиме стационарности.

После оценки модели по указанным выше параметрам, если свойства модели разработчика не удовлетворяют, выполняется ее **калибровка**. Это может быть, например, изменение типов событий, введение новых процессов, изменение законов распределения моделируемых входящих величин и др.

2.4.8. Планирование и проведение

имитационного эксперимента

Перед проведением рабочих расчетов на ЭВМ должен быть составлен план проведения эксперимента с моделью. При этом нужно четко представлять, какие параметры системы исследуются (зависимые), а какие параметры при этом нужно задавать (независимые). Далее определяется количество прогонов, необходимых для исследования системы. Затем определяется способ

получения результатов и их обработки. Чаще всего для получения результатов достаточно стандартной статистики.

После составления плана проведения эксперимента, можно приступить к выполнению рабочих расчетов и получению результатов моделирования. Моделирование рационально выполнять в два этапа: сначала контрольные, а затем рабочие расчеты. Контрольные расчеты проводятся для проверки машинной модели и определения чувствительности результатов к изменению исходных данных.

В связи с выдвинутой гипотезой о существенности сокращения времени обслуживания с 10 до 8 мин на одной из колонок, в рамках GPSS World предполагается использовать дисперсионный анализ ANOVA.

2.4.8.1. Эксперимент ANOVA (дисперсионный анализ)

Процедура используется для проведения дисперсионного анализа. Дисперсионный анализ позволяет определить, является ли изменение какого-либо параметра модели действительно значимым в изменении результатов, т.е., существенно ли это меняет итоговые характеристики, выдаваемые в отчете. Например, как увеличение скорости обслуживания транзакта в устройстве на определенную величину уменьшит время ожидания в очереди. Безусловно, это время уменьшится, но задачей дисперсионного анализа является определение того, будет ли это уменьшение значимым, или оно будет находиться в пределах случайных колебаний данной величины.

Сущность дисперсионного анализа состоит в разбиении общей дисперсии признака на три вида дисперсии: межгрупповую, внутригрупповую и остаточную:

$$\delta_{\text{общая}}^2 = \delta_{\text{межгрупповая}}^2 + \delta_{\text{внутригрупповая}}^2 + \delta_{\text{остаточная}}^2$$

Общая дисперсия признака отражает колебания индивидуальных значений признака вокруг среднего. Группы в данном случае отражают отклики (время нахождения в очереди) при разных значениях времени обслуживания. Например, имеется по три прогона модели с двумя разными значениями характеристики «время обслуживания в устройстве» 8 мин и 10 мин. Этот фактор влияет на показатель «среднее время нахождения транзакта в очереди». Значит, мы имеем две группы значений признака (например, первая группа – при скорости работы устройства – 10 мин., вторая – при скорости работы

устройства – 8 мин.) – «длина очереди» по 3 значения в каждой группе. Исчислив среднюю длину очереди по этим шести значениям, мы получим общую среднюю длину ($\bar{x}_{\text{общ}}$). Общая вариация признака ($W_{\text{общая}}$) рассчитывается так:

$$W_{\text{общая}} = \sum (y_i - \bar{y}_{\text{общ}})^2,$$

где y_i – i -е значение признака $\bar{y}_{\text{общ}}$ – среднее значение по 6 цифрам, общее число наблюдений (6).

Общая вариация складывается из трех:

1) *Межгрупповая вариация* отражает колебания средних значений по группам ($\bar{y}_{\text{гр}}$) вокруг общей средней ($\bar{y}_{\text{общ}}$):

$$W_{\text{межгр}} = \sum (\bar{y}_{\text{гр}} - \bar{y}_{\text{общ}})^2$$

2) *Внутригрупповая вариация* отражает колебания индивидуальных значений (y_i) от внутригрупповых средних ($\bar{y}_{\text{гр}}$):

$$W_{\text{внутригр}} = \sum (y_{ik} - \bar{y}_{\text{гр}})^2,$$

где k – номер группы.

3) Остаточная вариация равна:

$$W_{\text{остаточная}} = W_{\text{общая}} - (W_{\text{межгрупповая}} + W_{\text{внутригрупповая}})$$

Решение о существенности влияния выносится на основе расчета статистического F -критерия (критерий F -Фишера). F -критерий оценивает значимость различий двух дисперсий (или вариаций) – остаточной и факторной (межгрупповой).

Общая методика состоит из следующих этапов:

1) Выдвижение нулевой гипотезы (H_0) о несущественности влияния фактора на отклик модели.

2) Расчет фактического значения F -критерия по формуле:

$$F = \frac{W_{\text{факторная}}}{W_{\text{остаточная}}} \quad \text{или} \quad F = \frac{W_{\text{остаточная}}}{W_{\text{факторная}}} \quad (\text{в числителе должна стоять}$$

большая из дисперсий)

3). По специальным таблицам F -распределения находят критическое значение F -критерия.

4). Сравниваются $F_{\text{фактическое}}$ и $F_{\text{табличное}}$. Если $F_{\text{фактическое}} \geq F_{\text{табличное}}$, то H_0 отвергается, если наоборот, то принимается.

Данный алгоритм является встроенной процедурой в систему GPSS World. Это – процедура ANOVA (analyze of variation). Задача исследователя состоит в формировании матрицы, содержащей по несколько значений Y , соответствующих каждому значению X .

В нашем случае, мы выдвигаем гипотезу, которая формулируется следующим образом: различие в длине очереди при разной скорости обслуживания незначительно. Обслуживание клиентов на заправке на колонке № 1 (например) занимает в среднем 10 мин и это образует очередь на обслуживание. Существенно ли изменится очередь, если внедрить новое устройство или увеличить скорость обслуживания, например, до 8 мин?

Поскольку время поступления транзактов в систему является величиной случайной, то различные прогоны модели со скоростью обслуживания автомобиля 10 мин дадут различные показатели среднего времени ожидания транзакта в очереди. Так же как и в случае, если время обслуживания равно 8 мин.

При использовании стандартной процедуры ANOVA для формирования групп к исходному файлу программы присоединяем новый текстовый файл text_AZS.txt.

; GPSS World Sample File – AZS.GPS

RESULTS MATRIX ,2,3 ;

Cut_Time EQU 10

Treatment EQU 1

RMULT 411

Start 100,NP

MSAVEVALUE RESULTS,1,1,QT\$ZAPRAVKA

Clear Off

RMULT 421

Start 100,NP

MSAVEVALUE RESULTS,1,2,QT\$ZAPRAVKA

Clear Off
RMULT 431
Start 100,NP
MSAVEVALUE RESULTS,1,3,QT\$ZAPRAVKA

Clear Off
Cut_Time EQU 8
Treatment EQU 2
RMULT 411
Start 100,NP
MSAVEVALUE RESULTS,2,1,QT\$ZAPRAVKA

Clear Off
RMULT 421
Start 100,NP
MSAVEVALUE RESULTS,2,2,QT\$ZAPRAVKA

Clear Off
RMULT 431
Start 100,NP
MSAVEVALUE RESULTS,2,3,QT\$ZAPRAVKA

Этот файл определяет глобальную матрицу (2×3) результатов, в которой будут храниться числовые значения – результаты прогонов для проведения дисперсионного анализа. В данном файле команда EQU используется для установления среднего времени заправки. Оператор RMULT задает разные значения генератора случайных чисел. Блок CLEAR OFF «сбрасывает» статистику между прогонами для предотвращения влияния предыдущих результатов на последующие. Операнд NP в блоке START отключает создание стандартного отчета о результатах моделирования.

Порядок выполнения процедуры ANOVA:

*1. После прогона модели (**Command/Create Simulation**) откройте окно «Journal». Далее необходимо присоединить текстовый файл text_AZS.txt. Это делается так:*

Command/Custom...(Команда/ввести)

2. В открывшемся окне наберите:

INCLUDE "text_AZS.TXT" <OK>.

При этом в окне журнала отражается время каждого прогона.

3. Для просмотра матрицы результатов, выберите **Window/ Simulation Window/Matrix Window** (Рисунок 22).

RESULTS			
Dim 1		Dim 2	
		1	2
1		19.220	18.380
2		18.957	18.390

Рисунок 22 – Матрица результатов серии прогонов

4. Для проведения дисперсионного анализа по этим данным выберите **Command/Custom...(Команда/ввести)** и в открывшемся окне наберите **SHOW ANOVA (Results,2,1) <OK>**.

Появится таблица результатов (Рисунок 23).

№4.9.sim - JOURNAL						
10/30/04 11:28:22	SHOW ANOVA (RESULTS,2,1)					
10/30/04 11:28:22						
10/30/04 11:28:22	ANOVA					
10/30/04 11:28:22						
10/30/04 11:28:22	Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square	F	Critical Value of F (p=.05)
10/30/04 11:28:22	A	0.011	1	0.011	0.045	7.71
10/30/04 11:28:22	Error	0.962	4	0.240		
10/30/04 11:28:22	Total	0.972	5			
10/30/04 11:28:22						
10/30/04 11:28:22	Treatment Level A	Count	Mean	Minimum	Maximum	95% C.I. (SE)
10/30/04 11:28:22	1	3	18.972	18.380	19.314	[18.186, 19.757]
10/30/04 11:28:22	2	3	18.887	18.391	19.313	[18.101, 19.673]
10/30/04 11:28:22						
10/30/04 11:28:22	0.4903009					

Рисунок 23 – Таблица результатов дисперсионного анализа

В данном конкретном случае в окне дисперсионного анализа $F_{\text{табл.}} = 7,71$, а $F_{\text{факт.}} = 0,4903009$. Следовательно, гипотеза о несущественности изменения длины очереди вследствие увеличения скорости обслуживания до 8 минут принимается, т.е., если произвести замену оборудования на более производительное, обслуживающее автомобиль за 8 минут, то это существенно не повлияет на длину очереди.

Следовательно, для существенного уменьшения очереди необходимо сокращать время обслуживания более высокими темпами и вводить более совершенное оборудование.

2.4.8.2. Оптимизационный эксперимент

Иногда постановка имитационной задачи предполагает нахождение оптимального значения какого-либо параметра системы или сочетания набора ее элементов. Для характеристики этих экспериментов будем оперировать следующими общепринятыми терминами и понятиями.

Факторы – это входные переменные (или *входы*), от которых зависят результирующие показатели работы моделируемой системы. Отклик (выход) – это выходные переменные модели, они являются зависимыми. Как правило, факторы (или факторные признаки) обозначают x , а отклики – y . Например, в системе «сельскохозяйственное предприятие» факторами являются такие характеристики как количество вносимых удобрений, соотношение средств производства и рабочей силы, количество солнечной радиации и др. Откликами являются такие характеристики работы системы как выход валовой продукции, прибыль, рентабельность и др.

Уровнями фактора называют различные значения, которые данный фактор может принимать. Понятно, что между уровнями фактора и откликами существует некая связь, отражающаяся *функцией отклика*:

$$y_l = f(x_1, x_2, x_3, \dots, x_i, \dots, x_m), \quad l = \overline{1, n}, \text{ где}$$

y_l – l -й отклик;

n – число анализируемых откликов;

x_i – i -й фактор;

m – число факторов.

Если функцию отклика представить в виде некой многомерной области, то она будет называться *поверхностью отклика*. Функция отклика формируется после проведения серии экспериментов с моделью и представляет собой уравнение регрессии, полученное методом наименьших квадратов.

На отклики системы может влиять множество факторов, которые также взаимодействуют между собой. Из множества факторов можно выделить наиболее существенные, используя процедуру отсеивающего эксперимента. Далее, если поставлена такая цель, находят экстремальные значения на поверхности отклика путем подъема по поверхности отклика к вершине. При

этом обычно используют метод скорейшего подъема – процедура пошагового подъема. Направление скорейшего подъема показывает относительные величины изменения факторов, обеспечивающих максимальное увеличение отклика. Поверхность отклика может иметь несколько максимумов.

Функция отклика обычно представляется в виде полинома первого порядка, реже – второго или третьего. Если это – полином первого порядка, то функция выглядит следующим образом:

$$y=a_0 + a_1x_1 + a_2x_2 + \dots + a_kx_k.$$

Для характеристики этих экспериментов будем оперировать следующими общепринятыми терминами и понятиями.

В этом случае используют встроенную в GPSS Word процедуру *оптимизирующего эксперимента*. Данный эксперимент предназначен для построения уравнения регрессии (поверхности отклика) для заданных факторов и поиска оптимального значения вычисляемого в модели значения (максимального или минимального).

В качестве примера возьмем задачу оптимизации числа работников цеха механосборки.

Пример. На производственном участке работает несколько человек. Чтобы изготовить изделие, работники сначала его собирают в течение 30 ± 5 мин, а потом обрабатывают с помощью специального устройства в течение 8 ± 2 мин. После обработки изделие считается готовым и работник приступает к сборке нового изделия. Прибыль от одного изделия составляет 5 единиц стоимости, использование машины обходится в 400 единиц стоимости в неделю, зарплата одного работника – 150 единиц стоимости в неделю.

Определить, сколько нужно рабочих, чтобы прибыль от производства была максимальной. Моделирование необходимо выполнить для пятидневной недели с восьмичасовым рабочим днем.

При единице модельного времени, равной одной минуте программа будет выглядеть следующим образом:

```
NWORKERS EQU 3
PRIBY FVARIABLE N$OUT#5-400-NWORKERS#150
GENERATE „,NWORKERS
BACK1 ADVANCE 30,5
```



```

SEIZE OVEN
ADVANCE 8,2
RELEASE OVEN
OUT      TRANSFER ,BACK1
GENERATE 2400
TERMINATE 1

```

Нужно найти такое значение параметра *NWorkers*, при котором доход достигал бы максимального значения. В выражение дохода *PRIBY* входит число изготовленных изделий, равное значению *СЧА N\$OUT*. При этом доход может быть как отрицательным (убытки от производства), так и положительным.

Процедура оптимизирующего эксперимента предполагает следующее. Оптимизирующий эксперимент добавляется в созданную и отлаженную модель (**Command/Create Simulation**). Для этого необходимо выбрать пункт меню **Edit/Insert Experiment/Optimizing**, после чего откроется диалоговое окно **Optimizing Experiment Generator** (Рисунок 24).

Initial Local Experimental Region			Movement Limits	
Factor Name (User Variable)	Value 1	Value 2	Low Limit	High Limit
A NWORKERS	1	16	0	100
B			0	100
C			0	100
D			0	100
E			0	100

Result
Expression: V\$PRIBY
☒ Maximize ☐ Minimize

☒ Generate Run Procedure ☒ Load F12 with CONDUCT Command

Insert Experiment Cancel Help

Рисунок 24 – Окно «Optimizing Experiment Generator»

В нем указываются необходимые для выполнения эксперимента параметры. В полях **Experiment Name** и **Run Procedure Name** задают соответственно название эксперимента и Run-процедуры прогонов модели. По умолчанию название эксперимента – **RSM** (от *Response Surface Model*). Это имя будет добавлено ко всем названиям процедур и переменных, которые автоматически генерируются для этого эксперимента. Пользователю доступны для изменений все процедуры, относящиеся к эксперименту, кроме одной – *RSM_FitSurfaceToData* (имя процедуры фиксировано и не изменится при задании другого названия эксперимента). В качестве названия Run-процедуры

рекомендуется использовать *DoTheRun*, хотя можно использовать и любое другое имя.

В полях **Factor Name A, B, C, D, E** указываются изменяемые величины (факторы) и задаются значения их верхних и нижних уровней **Value 1** и **Value 2**.

Поля **Low Limit, High Limit** для **Movement Limits** задают максимальные границы перемещения (могут иногда помочь при поиске оптимума, но в большинстве случаев можно оставлять значения по умолчанию 0 и 100).

Значение поля **Redirection Limit** ограничивает количество изменений направлений при поиске оптимума. В некоторых случаях достаточно 1, но обычно нужно выбирать 2 или 3.

В поле **Expression** указывают целевую функцию – выражение на языке PLUS, заключенное в круглые скобки, или переменная модели в которой задана эта функция. С помощью переключателя **Maximize/Minimize** выбирается направление оптимизации.

При выборе **Generate Run Procedure** создается процедура прогонов модели, в противном случае придется ее написать. Выбор **Load F12** позволяет запускать эксперимент с помощью нажатия одной клавиши **F12**.

Для запуска автоматической генерации эксперимента нужно нажать на кнопку **Insert Experiment**, но сначала необходимо задать перечисленные параметры.

После нажатия кнопки **Insert Experiment** откроется окно **Run Procedure Generation** (Рисунок 25 и 26), в котором отображается автоматически сгенерированный текст Run-процедуры, названной именем *DoTheRun*.

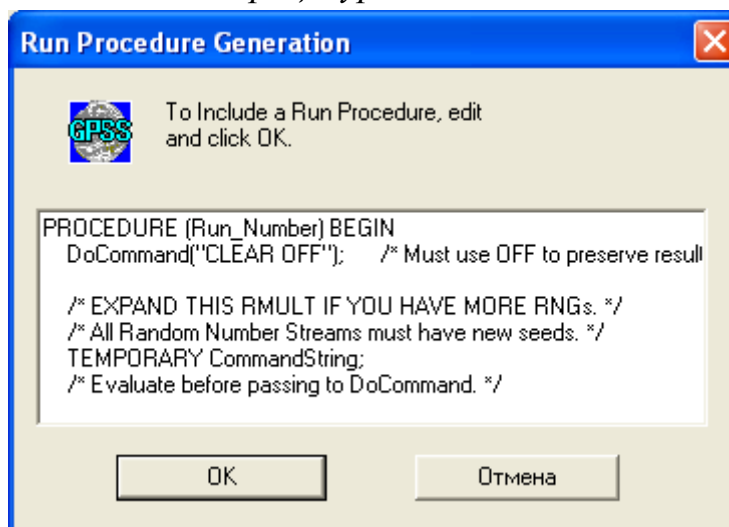


Рисунок 25 – Окно «Run Procedure Generation»

В него нужно внести исправления, в данном случае необходимо следующим образом изменить процедуру **DoTheRun**:

```

PROCEDURE DoTheRun(Run_Number) BEGIN
DoCommand("CLEAR OFF");
DoCommand ("START 1");
END;

```

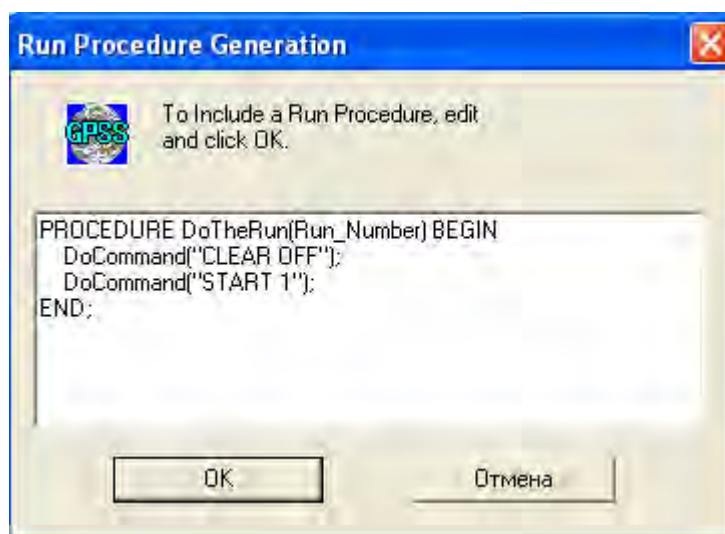


Рисунок 26 – Откорректированное окно «Run Procedure Generation»

Теперь надо оттранслировать модель, выполнив пункт меню **Command/Create Simulation**. При этом мы увидим, что все процедуры регистрируются.

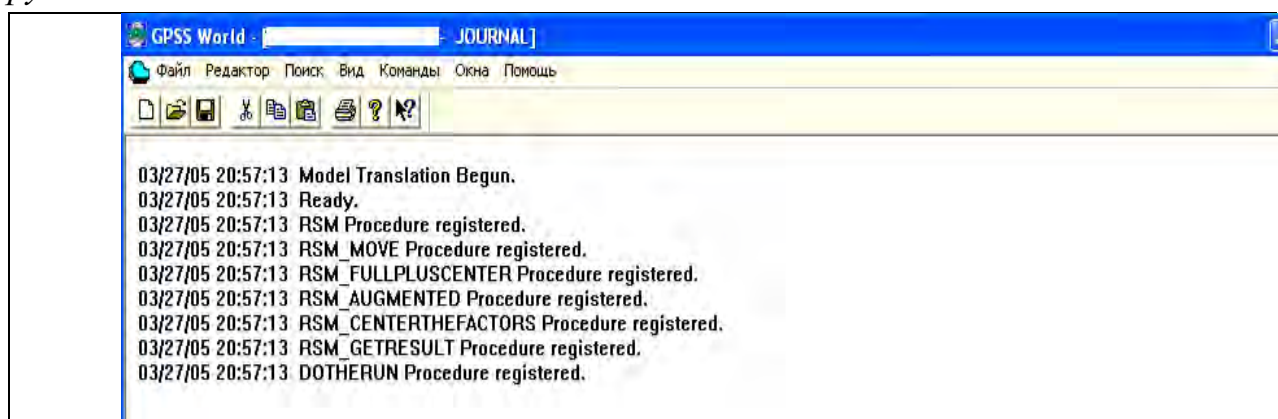


Рисунок 27 – Журнал регистрации процедур эксперимента

Нажатием клавиши **F12** запускают эксперимент. В журнале сессии (**JOURNAL**) будет представлена следующая информация о ходе эксперимента (Рисунок 28).

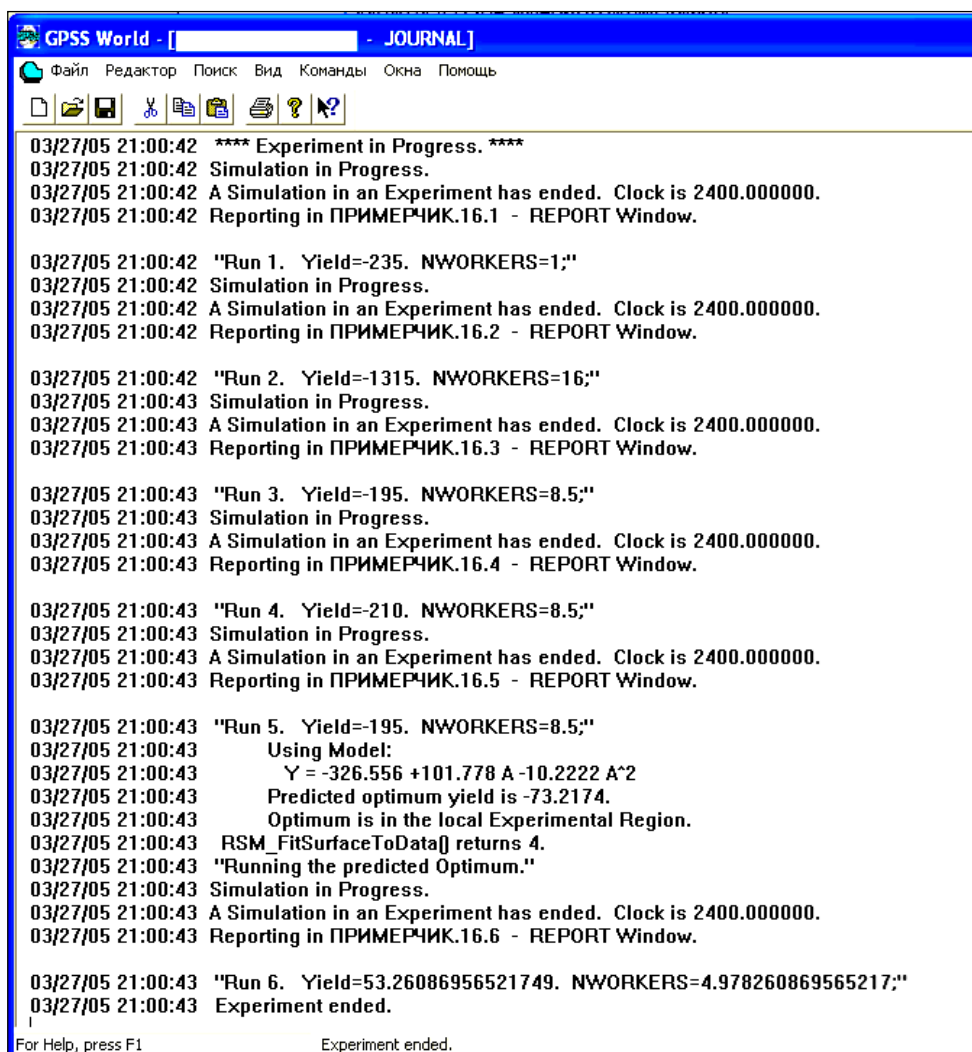


Рисунок 28 – Журнал оптимизационного эксперимента

Таким образом, получен следующий результат: в процессе поиска решения была построена модель второго порядка (уравнение отклика): $Y = -326,556 + 101,778A - 10,222A^2$.

Оптимальные значения: входная переменная $NWorkers = 4,987$, выходная переменная $Yield$ (прибыль) составляет 53,26 ед. С учетом условия целочисленности количества работников, имеем: оптимальное число работников для этой модели равно 5, а величина дохода при этом составит 53,26 ед.

Контрольные вопросы к главе II

1. Сущность систем массового обслуживания (СМО). Примеры.
2. Что такое транзакт (заявка)?
3. Что такое закон поступления заявок? Дисциплина выхода заявок из очереди? Правила обслуживания в устройстве.
4. Назовите основные характеристики работы СМО.

5. Сети систем массового обслуживания.
6. Общая характеристика системы имитационного моделирования GPSS World.
7. Какие блоки существуют в системе GPSS World?
8. Классификация объектов GPSS.
9. Какие объекты относятся к объектам типа «ресурс»?
10. Типы операторов в GPSS World.
11. Формат блоков в GPSS World
12. Блок GENERATE. Операнды блока.
13. Операторы занятия и освобождения обслуживающего устройства.
14. Оператор обслуживания в устройстве.
15. Оператор моделирования входа и выхода транзактов из очереди.
16. Оператор Переход транзакта в блок, отличный от последующего.
17. Операторы сохранения вычисляемых величин.
18. Блок TEST
19. Моделирование переменных величин в GPSS World
20. Понятие стандартного числового атрибута (СЧА).
21. Моделирование и использование функций.
22. Анализ и интерпретация результатов пробного прогона модели.
23. Перечислите свойства имитационной модели.
24. Что такое адекватность имитационной модели.
25. Понятие точности результатов моделирования.
26. Как определить необходимый объем выборки при оценке точности результатов моделирования?
27. Алгоритм проведения эксперимента ANOVA.
28. Оптимизационный эксперимент. Постановка задачи и оценка результатов эксперимента.

Задания для разработки имитационного проекта в системе GPSS World

Задание 1. В сельскохозяйственной организации работает ремонтный цех. На текущий ремонт поступают грузовые автомобили и трактора. Интервал их поступления и время обслуживания представлены в таблице:

Вид техники	Интервал поступления, мин	Время обслуживания, мин
Грузовые автомобили	55 ± 15	28 ± 5
Трактора	35 ± 10	30 ± 8

Какова себестоимость произведенной работы за время моделирования, если себестоимость ремонта автомобиля – 1 500 р., а трактора – 2 000 р.

На текущий ремонт также поступает транспорт главных специалистов хозяйства, который обслуживается вне очереди. Интервал поступления составляет 180 ± 20 минут, время обслуживания 30 ± 5 мин. Необходимо смоделировать работу участка в течение 40 ч. Определить длину очереди на текущий ремонт и среднее время ожидания обслуживания. Внести предложения по сокращению времени обслуживания в системе.

Оценить статистически, существенно ли уменьшится длина очереди грузовых автомобилей, если время их обслуживания сократится с 28 до 20 мин.

Задание 2. В информационном центре производят обработку заказов на реализацию. Принимаются три класса заказов: розничные, мелкооптовые и крупнооптовые. Розничные заказы поступают через 20 ± 5 мин, мелкооптовые – через 20 ± 10 мин и крупнооптовые заказы – через 50 ± 10 мин. Продолжительность оформления: розничные заказы – 20 ± 5 мин, мелкооптовые – 21 ± 3 мин и крупнооптовые – 28 ± 5 мин.

Исходя из наличия функциональных возможностей ЭВМ, заказы по рознице и мелкому опту принимаются и обрабатываются одновременно, а крупнооптовые заказы выполняются вне очереди.

Смоделировать работу информационного центра за 80 ч. Определить загрузку Центра.

Если крупнооптовые заказы будут комплектовать два работника, и затрачивать на это не 28, а 25 мин, то, существенно ли это сократит длину очереди (оценить статистически).

Задание 3. На ремонтный участок МТС, который состоит из 2-х боксов (для ремонта и для техосмотра) через каждые 20 ± 3 минуты поступают тракторы. Из них 60 % требуют ремонта, который продолжается 40 ± 15 минут, 40 % тракторов проходят технический осмотр продолжительностью 20 ± 5 мин. Смоделировать работу участка в течение 100 ч. Определить вероятность отказа в обслуживании. Рассмотреть основные характеристики работы системы. Предложить меры по оптимизации обслуживания техники, используя инструмент ANOVA.

Какова общая стоимость выполненной работы, если стоимость ремонта составляет 200 р., а техосмотра – 50 р.

Задание 4. На участке происходит сортировка и упаковка яблок, которые поступают в ящиках через 20 ± 5 мин. Сортировка одного ящика занимает 10 ± 7 мин, а упаковка – 7 ± 5 мин. Сортировкой занимается один ра-

ботник, а упаковкой – другой. Известно среднее время для передачи яблоч от одного работника к другому – 2 ± 1 мин.

Смоделировать процесс работы участка в течение 8 ч. Определить длину очереди на этапе сортировки и упаковки.

Внести предложения по сокращению длины очереди, т.е. насколько должно уменьшиться время обслуживания, чтобы это значительно сократило очередь (провести несколько экспериментов ANOVA)?

Какова дневная зарплата работников, если за один отсортированный ящик рабочие получают 30 р.

Задание 5. На складе ремонтного предприятия работают 2 кладовщика. Один из кладовщиков (первый) выполняет работу по поиску и выдаче запасных частей немного быстрее. Поэтому работники отдают предпочтение первому кладовщику. Работники предприятия заходят на склад для поиска и получения запасных частей каждые 7 ± 2 минуты. Первый кладовщик обслуживает работника за 3 ± 1 минуты, второй за 5 ± 1 минуты. Промоделируйте работу склада на протяжении 40 ч. Определите длину очереди и время ожидания обслуживания к каждому кладовщику. Насколько существенной для результатов будет замена второго кладовщика на работающего более быстро и выполняющего заказ не за 5, а за 4 мин? Оценить статистически.

Задание 6. Диспетчер управляет внутризаводским транспортом и имеет в своем распоряжении два грузовика. Заявки на перевозки поступают к диспетчеру каждые 5 ± 4 мин. Диспетчер запрашивает по радио один из грузовиков и передает ему заявку, если тот свободен. В противном случае он запрашивает другой грузовик и таким образом продолжает сеансы связи, пока один из грузовиков не освободится. Диспетчер допускает накопление у себя до пяти заявок, после чего вновь прибывшие заявки получают отказ. Грузовики выполняют заявки на перевозку за 12 ± 8 мин.

Смоделировать работу внутризаводского транспорта в течение 10 ч. Подсчитать число обслуженных и отклоненных заявок. Определить коэффициенты загрузки грузовиков.

Оценить статистически целесообразность замены грузовиков на более производительные, выполняющие перевозку за 8 мин.

Задание 7. В ремонтную мастерскую тракторов ООО «Салют» на капитальный ремонт поступают тракторы каждые 50 ± 10 дней, ремонт длится 20 ± 5 дней. Текущий ремонт (замена и ремонт топливного насоса, замена свечей, колес и т.д.) производится за $1,5 \pm 0,5$ дня. Необходимость в текущем ремонте возникает в среднем каждые 10 ± 3 дня. В мастерскую поступают также машины работников аппарата управления хозяйством через 15 ± 5

дней, которые имеют более высокий приоритет обслуживания, чем капитальный ремонт, но ниже, чем текущий. Их ремонт производится обычно за 5 ± 1 день. Необходимо:

- 1) Смоделировать работу мастерской в течение полугода.
- 2) Определить коэффициент использования мастерской.

Мастерская функционирует по принципу хозрасчета. Если капитальный ремонт трактора дает чистый доход в размере 5 000 р., а текущий – 3 000 ., то каков будет чистый доход от проведения ремонта за период моделирования?

Если покупка нового оборудования для ремонта увеличит скорость текущего ремонта тракторов с 1,5 до 0,7 дня, существенно ли это отразится на уменьшении очереди? Оценить статистически.

Задание 8. В картофелехранилище процесс сортировки картофеля организован следующим образом: через 10 ± 5 мин на сортировочный пункт, где работает два сортировщика, поступает два мешка картофеля на переборку. В целях ведения индивидуального учета выполненной работы каждый работник перебирает отдельный мешок. Первый работник сортирует содержимое одного мешка за 15 ± 3 мин, второй – за 18 ± 5 мин. Если первый работник занят переборкой во время поступления очередной партии мешков, то мешки поступают ко второму работнику (если он уже освободился).

Определить коэффициент использования первого и второго работника и число отсортированных ими мешков картофеля.

Если выручка от реализации одного мешка картофеля (50 кг) составляет 3 500 р., а себестоимость производства и реализации 1 кг картофеля (включая сортировку), составляет 4 р., то какова прибыль предприятия от реализации картофеля за данный день (рассчитать по данным стандартного отчета).

Если второго рабочего заменить на рабочего, выполняющего эту работу за 8 минут, существенно ли это повлияет на общую скорость их работы и уменьшит ли число мешков, принесенных в хранилище и требующих переборки? Оценить статистически.

Задание 9. В пекарню, где работает два работника, через 20 ± 5 минут поступает партия (мешок) муки. Если в момент поступления мешка первый работник, изготавливающий кондитерские изделия за 40 ± 7 мин, занят, то мешок берет второй работник, перерабатывающий его за 30 ± 5 мин. Промоделировать работу цеха в течение 8 ч.

Определить коэффициент использования первого и второго работника и число использованных ими упаковок с мукой.

Если выручка от реализации кондитерских изделий, произведенных из одной упаковки, составляет 200 р., а себестоимость их производства и реализации составляет 150 р., то какова прибыль предприятия от реализации картофеля за данный день (рассчитать по данным стандартного отчета).

Если первого работника обязать работать быстрее и выполнять эту операцию за 32 ± 5 минут, существенно ли это повлияет на общую скорость их работы и существенно ли уменьшит очередь? Оценить статистически.

Задание 10. Частота поломок компьютеров сети подчиняется пуассоновскому потоку интенсивностью $\lambda = 0$ и $\beta = 20$. Сломавшийся компьютер немедленно ремонтируется. Время ремонта компьютера составляет 30 ± 10 мин. Промоделировать работу системы в течение 8-и часового рабочего дня.

Получить стандартный отчет о результатах моделирования.

Определить:

- а) коэффициент использования компьютеров;
- б) среднее время ремонта компьютера;
- в) количество обращений к мастеру в течение смены.

Если нанять мастера по ремонту компьютеров, ремонтирующего один компьютер за 20 мин, а не за 30 мин, как предыдущий, существенно ли это отразится на результатах работы системы? Оценить статистически.

Если ремонт одного компьютера обходится 200 р., то какова стоимость произведенного за смену ремонта?

Задание 11. В ремонтной мастерской изготавливают гайки диаметром 10 и 12 мм, используя при этом три разных станка. Интервалы поступления в мастерскую заготовок для изготовления гаек первого вида составляют 50 ± 7 мин, а заготовок для гаек второго вида – 25 ± 4 мин. Время обработки этих заготовок следующее:

Вид работы	Гайки 1-го вида	Гайки 2-го вида
Резка	16 ± 4	18 ± 3
Нарезка резьбы	25 ± 5	27 ± 5
Упаковка	20 ± 3	21 ± 4

Смоделировать работу мастерской за 8-часовой рабочий день при двухсменном режиме

Определить:

- 1) среднюю загрузку каждого станка;
- 2) среднее время изготовления детали каждого вида;
- 3) длину очереди перед каждым станком;
- 4) объем упаковочной тары для каждого вида гаек (шт.).

Если чистый доход от реализации гайки первого вида составляет

0,8 р., а второго вида – 1,2 р., то каков чистый доход за год при работе в данном режиме.

Каково будет управленческое решение по возможной замене машины, производящей упаковку гаек второго вида, что увеличит скорость упаковки с 21 мин до 18 мин (используя статистические критерии).

Задание 12. В плодоовощном цехе производится мойка яблок и груш и их упаковка в пластиковые контейнеры для последующей реализации. При этом используются моечная и упаковочная машины. Интервалы поступления в цех ящиков с яблоками составляет 10 ± 5 мин, а ящиков с грушами – 15 ± 4 мин. Время мойки и упаковка яблок и груш составляет:

Вид работы	Яблоки	Груши
Мойка	10 ± 1	12 ± 3
Упаковка	5 ± 2	7 ± 3

Смоделировать работу цеха за 8-часовой рабочий день.

Определить:

- 1) среднее время мойки и упаковки фруктов каждого вида;
- 2) среднюю загрузку моечной и упаковочной машин;
- 3) длину очереди перед каждой машиной;
- 4) объем тары для перевозки каждого вида фруктов (ящиков).

Если прибыль от реализации одного контейнера с яблоками составляет 30 р., а от реализации груш – 50 р., то какова дневная прибыль предприятия?

Каково будет управленческое решение по возможной замене машины, производящей мойку груш, что увеличит скорость мойки с 12 до 7 мин (использовать статистические критерии).

Задание 13. В период заготовки овощей на овощную базу хозяйства поступают грузовые автомобили с овощами, которые проходят процедуру взвешивания. В связи с напряженным графиком работы на базе открыто две весовые. Поток автомобилей, поступающих на эти весовые, подчиняется экспоненциальному закону распределения вероятностей с параметрами $\lambda = 0$ и $\beta = 7,1$. Время взвешивания на первых весах составляет 8 ± 3 мин, на вторых весах – 10 ± 4 мин. Машина подъезжает к свободным весам.

Смоделируйте работу весовой в течение рабочей смены (8 ч).

Определить:

- 1) коэффициент загрузки каждой из весов;
- 2) среднее время взвешивания на первых и вторых весах;
- 3) максимальное, среднее и текущее число автомобилей в очереди к весам;
- 4) среднее время нахождения автомобиля в очереди.

Подсчитано, что условный убыток от простоя одного автомобиля в течение 1 мин составляет 5 р., то каков условный убыток от простоя автомобилей в очереди к весам за одну смену?

Если заменить вторые весы на более производительные, взвешивающие за 7 минут, то существенно ли это повлияет на среднее время нахождения автомобилей в каждой очереди (оценить статистически).

Задание 14. На молочный завод из хозяйств приезжают молоковозы, сдающие молоко на анализ в лабораторию завода. На заводе имеется две лаборатории. Поток молоковозов, поступающих на завод, подчиняется экспоненциальному закону распределения вероятностей с параметрами $\lambda = 0$ и $\beta = 6,8$. Время анализа молока в первой лаборатории, имеющей специальное оборудование, составляет 15 ± 3 мин, во второй лаборатории – 20 ± 4 мин. Машина подъезжает к освободившейся лаборатории.

Смоделируйте работу по анализу молока в течение рабочей смены (8 ч).

Определить:

- 1) среднее время нахождения молоковоза в очереди;
- 2) среднее время анализа в первой и второй лабораториях;
- 3) максимальное, среднее и текущее число молоковозов в очереди;
- 4) коэффициент загрузки каждого весов.

Если во второй лаборатории установить оборудование, которое будет производить анализ за 15 мин, то существенно ли это уменьшит длину очереди (эксперимент ANOVA)?

Подсчитано, что условный убыток от простоя одного молоковоза в течение 1 мин составляет 7 р. Каков условный убыток в хозяйстве от простоя молоковоза в очереди к лабораториям за одну смену?

Задание 15. В банк приходят клиенты для оформления оплаты за коммунальные услуги. В расчетном зале работает два оператора по оформлению данных операций. Поток посетителей банка подчиняется равномерному закону распределения с интенсивностью 3 ± 1 мин. Первый оператор производит расчет быстрее – за 8 ± 2 мин., второй – за 10 ± 3 мин. Клиент подходит к свободному оператору.

Смоделируйте работу банка в течение рабочей смены (8 ч).

Определить:

- 1) коэффициент загрузки каждого из операторов;
- 2) среднее время обслуживания клиентов первым и вторым операторами;
- 3) максимальное, среднее и текущее число клиентов в очереди;

4) среднее время нахождения клиента в очереди.

Если за обслуживание одного клиента банк берет комиссионные 10 р., то какова сумма комиссионных, полученных банком за день?

Если второго оператора заменить автоматом, который будет проводить расчет за 4 мин, существенно ли это повлияет на сокращение очередей в расчетном зале банка (провести эксперимент ANOVA).

Задание 16. На склад запчастей в хозяйстве каждые 60 ± 10 минут приходят трактористы за запчастями. Тракторист может взять одну или несколько запчастей для трактора. Склад имеет два помещения, в которых хранятся различные запчасти. Время поиска необходимой запчасти в первом помещении составляет 8 ± 3 мин, во втором 10 ± 4 мин. Вероятности поиска запчастей в данных помещениях соответственно равны 0,55 и 0,65. Количество запчастей, взятых из этих помещений, составляет: из первого помещения – 5 ± 2 , из второго – 4 ± 1 , из третьего – 3 ± 1 .

После нахождения необходимых запчастей тракторист подходит с ними к кладовщику и делает соответствующую запись в книге учета. Чем больше запчастей взял тракторист, тем больше занимает время оформления. На оформление одной запчасти требуется 2 мин. Определить:

- 1) количество трактористов, посетивших склад в течение 8-часовой смены;
- 2) среднее время поиска запчасти;
- 3) коэффициент использования кладовщика;
- 4) среднюю длину очереди на складе;
- 5) число входов на склад без ожидания;
- 6) среднее время оформления взятия запчасти.

Если в обязанности кладовщика вменить поиск запчастей, то скорость нахождения запчастей увеличится по первому помещению, где находится кладовщик, до 5 мин. Целесообразно ли принимать подобное управленческое решение. Оценить статистически.

Определить себестоимость услуг кладовщика за день, если себестоимость оформления одной детали составляет 10 р.

Задание 17. Смоделировать приход для обслуживания в банке 500 клиентов. В банк по одному приходят клиенты в среднем через каждые 10 мин. Интервал отклонения от среднего времени прихода подчиняется экспоненциальному закону распределения. В зале находится три оператора для ведения расчетов. Клиент подходит к незанятому окошку, в противном случае он ждет своей очереди в специально отведенной комнате. Операция обслужива-

ния клиентов занимает в среднем 6 ± 2 мин. После этого клиент покидает помещение расчетного центра.

Определить:

- 1) коэффициент использования каждого канала обслуживания;
- 2) среднее время использования каждого канала обслуживания;
- 3) количество транзактов, вошедших в каждый канал обслуживания;
- 4) среднее содержимое накопителя;
- 5) среднее время пребывания клиента в накопителе;
- 6) максимальное содержимое накопителя;
- 7) коэффициент использования накопителя.

Если банк берет комиссионные в размере 10 р. с человека, то каков комиссионный сбор в течение дня?

Если ввести автоматизированные расчеты с временем оформления операции за 2 мин, то существенно ли это уменьшит содержимое накопителя? Оценить статистически.

Задание 18. Смоделировать работу расчетного отдела коммерческого банка в течение 8-часовой смены. В расчетный отдел коммерческого банка поступают платежные поручения. Среднее время поступления равно 15 мин. Отклонения в их поступлении от среднего времени поступления подчиняется экспоненциальному закону распределения вероятностей с параметрами $\lambda = 0$, $\beta = 5,4$. При их поступлении служащий банка берет документ и направляется в расчетный отдел, в котором платежное поручение обрабатывается за 6 ± 2 мин. В расчетном отделе работает три операциониста.

Определить:

- 1) коэффициент использования каждого канала обслуживания;
- 2) среднее время использования каждого канала обслуживания;
- 3) количество входов в каждый канал обслуживания.

Если банк берет комиссионные в размере 12 р. за оформление одного платежного поручения, то каков комиссионный сбор в течение дня?

Оценить статистически, целесообразно ли привлечение в расчетный отдел работника, который будет обрабатывать заявки со скоростью 4 мин.

Задание 19. Смоделировать работу двух бухгалтеров хозяйства, выполняющих расчеты по накладным. В бухгалтерию хозяйства поступают накладные в среднем через каждые 15 мин с интенсивностью поступления, аналогичной экспоненциальному закону распределения вероятностей ($\lambda = 0$, $\beta = 3,5$). Обработка накладной производится каждым из бухгалтеров за 30 ± 5 мин. Время моделирования – один месяц (22 рабочих дня при 8-часовом рабочем дне).

Определить:

- 1) коэффициент загрузки каждого бухгалтера;
- 2) среднее время использования каждого канала обслуживания (бухгалтера);
- 3) число накладных, обработанных каждым из бухгалтеров.

Если условная стоимость обработки одной накладной составляет 0,3 р., то какова стоимость выполненной работы за месяц?

Если ввести автоматизированную обработку накладных с помощью компьютера, когда скорость обработки увеличится с 30 до 20 мин, существенно ли это изменит число обработанных накладных? Оценить статистически.

Задание 20. Смоделировать работу цеха по производству мяса птицы. В специальное помещение для приема входит 50 кур. Поток тушек является равномерным с интервалом поступления 5 ± 1 мин. На первом участке цеха, где производится ошпаривание тушек, операция с одной тушкой длится $1 \pm 0,4$ мин., второй участок цеха производит ощипывание (одна тушка – в течение 10 ± 5 мин.). Затем тушку удаляют из цеха.

Смоделировать работу цеха в течение 8 ч.

Получить стандартный отчет о результатах моделирования.

Определить:

- коэффициент загрузки цеха;
- максимальное, среднее и текущее количество тушек в цехе;
- среднее время пребывания одной тушки в цехе.

Если операция по ощипыванию тушек будет проводиться с помощью специальной машины, то это увеличит скорость ощипывания с 10 до 5 мин. Существенно ли это сократит длину очереди? Оценить статистически.

Если себестоимость обработки в цехе одной тушки составляет 7 р., то какова себестоимость проведенных работ за 8-часовую смену.

Задание 21. Работники цеха хлебобулочных изделий делают заготовки из теста для последующей выпечки булок. Одну партию изделий изготавливают за 40 ± 7 мин и относят в печь. Выпечка производится в течение 15 ± 4 мин. Цена одного изделия составляет 7 р. Использование печи обходится предприятию в 800 р. в неделю. Зарплата одного работника оставляет 100 р. в час.

Получить и проанализировать стандартный отчет о результатах моделирования. Определить оптимальное с точки зрения максимума прибыли количество работников в цехе.

Моделирование выполнить для 5-дневной рабочей недели с часовым рабочим днем.

8-

Задание 22. В мясном цехе работает два работника. В цех с интервалом 30 ± 5 минут поступает туша коровы. Если в момент поступления туши первый работник занят, то ее начинает разделывать второй работник. Первый работник разделывает тушу за 40 ± 5 мин, второй – за 35 ± 5 мин. Смоделировать работу в течение 5 ч.

Получить и проанализировать стандартный отчет о результатах моделирования.

Определить коэффициент загрузки первого и второго работника и количество разделанных ими туш животных.

Если выручка от реализации одной туши составляет 10 000 р., а себестоимость их производства и реализации составляет 7 500 р., то какова прибыль предприятия от реализации мяса за день?

Если первый работник будет заменен на работника, выполняющего разделку не за 40, а за 35 мин, существенно ли это уменьшит очередь? Оценить статистически.

Задание 23. Смоделировать работу машинно-тракторной станции (МТС) в течение одного месяца. Заявки на механизированное обслуживание поступают в среднем 1 раз в три дня. Интенсивность их поступления подчиняется экспоненциальному закону распределения вероятностей. При поступлении заявок их ставят в очередь. Освободившаяся техника обслуживает одну заявку в среднем за 5 ± 2 дня. На МТС имеется две бригады, работающие параллельно.

Определить:

- 1) коэффициент использования каждой бригады;
- 2) среднее время работы каждой бригады;
- 3) количество заявок, выполненных каждой бригадой;
- 4) среднее число заявок в очереди;
- 5) среднее время пребывания заявки в очереди;
- 6) максимальную длину очереди.

Если стоимость работ составляет в среднем 5 тыс. р. на одну заявку, то какова стоимость выполненной работы МТС за период моделирования?

Оценить статистически целесообразность привлечение дополнительных работников в первую бригаду, чтобы увеличить скорость ремонта с 5 до 4 дней.

Задание 24. В сельскохозяйственной организации работает ремонтный участок. На текущий ремонт поступают сельскохозяйственные машины и

оборудование животноводческих ферм. Их характеристики представлены в таблице:

Вид техники	Интервал поступления, мин	Время обслуживания, мин
Сельскохозяйственные машины	40 ± 15	28 ± 5
Оборудование для ферм	90 ± 10	50 ± 7

Поступление на ремонт оборудования для ферм прерывает работу участка и имеет более высокий приоритет в связи с хозяйственной необходимостью.

Необходимо смоделировать работу участка в течение 100 ч. Определить длину очереди на ремонт и время ожидания на обслуживание. Внести предложения по сокращению времени обслуживания в системе.

Оценить статистически существенно ли уменьшится длина очереди сельскохозяйственных машин, если время их обслуживания сократится с 28 до 20 мин.

Определить хозрасчетный доход данного подразделения за это время, если доход от ремонта одной сельхозмашины составляет 500 р., а единицы оборудования для ферм – 1 500 р.

Задача 25. К рабочим для изготовления детали поступают заготовки с транспортного конвейера. Интервал между поступлениями двух идущих одна за другой деталей равен 4 ± 1 мин. Время изготовления детали первым рабочим составляет 12 ± 1 мин, а вторым – 13 ± 2 мин. Если первый рабочий занят, заготовка с конвейера перемещается ко второму рабочему. Требуется смоделировать работу линии в течение смены. Необходимо определить число деталей, обработанных каждым рабочим за смену, коэффициент загрузки рабочих, максимальную и среднюю длину очередей деталей и среднее время пребывания детали в очереди к третьему рабочему.

Оценить статистически существенно ли уменьшится длина очереди, если рабочих заменить автоматом, выполняющим работу не за 35 мин, как рабочие, а за 20 мин.

Задача 26. На склад элеватора поступает зерно яровой пшеницы. Подвоз зерна на склад осуществляют автомобили ГАЗ-53. Время прихода автомобиля распределено равномерно в пределах 20 ± 10 минут. Перед разгрузкой автомобиля, зерно осматривает один из 2-х селекционеров. Время осмотра первым селекционером составляет 25 ± 7 , а вторым – 30 ± 5 мин. После осмотра зерну присваивают определенный класс, его разгружают в соответствующем помещении склада. Если первый селекционер занят, оценку пар-

тии проводит второй.

Требуется смоделировать работу первого и второго селекционеров в течение смены. Необходимо определить коэффициент использования первого и второго селекционеров (первого и второго каналов обслуживания), число оцененных партий зерна, выполненных каждым из них, максимальную и среднюю длину очереди на обслуживание. При моделировании учитывать, что одна партия – это одна машина с зерном.

Оценить статистически существенно ли уменьшится длина очереди грузовых автомобилей, если время осмотра автомобиля первым селекционером сократится с 25 до 15 мин.

Задача 27. Овощеводческое звено плодоовощного хозяйства производит переборку и калибровку овощей. Ящики с овощами подают с интервалом 7 ± 2 мин. Сортировкой овощей заняты 4 работницы звена. Время переборки овощей первой работницей составляет 12 ± 1 мин, второй – 10 ± 2 мин, третьей – 8 ± 2 , четвертой – 15 ± 4 . Требуется смоделировать работу работниц в течение смены. Необходимо определить коэффициент использования каждой работницы и число перебранных ящиков каждой из них. Определить насколько существенно изменит количество отсортированных ящиков овощей привлечение дополнительной работницы.

Оценить статистически существенно ли уменьшится длина очереди ящиков, если первая работница будет перебирать ящик за 8 мин.

Задача 28. К рабочим лакокрасочного цеха поступают на покраску изделия деревообрабатывающего цеха. Интервал между поступлениями двух идущих друг за другом изделий составляет 5 ± 2 мин. Время покраски изделия первым рабочим составляет 10 ± 1 мин, а вторым – 12 ± 2 мин, третьим – 15 ± 2 . Если первый рабочий занят, деталь перемещается ко второму рабочему, если и он занят – к третьему. Требуется смоделировать работу всей линии в течение смены. Необходимо определить число изделий, обработанных каждым рабочим за смену, коэффициент загрузки рабочих, максимальную и среднюю длину очередей и среднее время пребывания транзактов в очередях.

Оценить статистически, существенно ли уменьшится длина очереди изделий, если время их покраски третьим рабочим сократится с 15 до 10 мин.

Задача 29. Производство концентрированных кормов в хозяйстве можно представить как многофазный процесс. На первом этапе осуществляется выбор и погрузка партии зерна, предназначенного для кормовых целей. Погрузка занимает 50 ± 10 минут для 3 т зерна. Далее – подвоз зерна к кормоцеху (занимает 15 ± 5). Разгрузка – 5 ± 1 мин. Переработка зерна (3 т) на комбикорм составляет $2 \pm 0,2$ ч. Смоделировать производство кормов в течение

3-х рабочих дней при условии, что подвоз кормов осуществляется автомобилями ГАЗ–53, которые подъезжают к складу с интервалом 50 ± 5 мин (грузоподъемность ГАЗ-53 составляет 3 т). Требуется определить основные характеристики системы:

- коэффициенты использования всех узлов обслуживания;
- среднее время пребывания партии зерна в очереди;
- среднее время пребывания автомобилей в очереди на загрузку;
- общее число входов в очередь, то есть общее число поступлений партий зерна на обслуживание в течение смены.

Оценить статистически существенно ли уменьшится длина очереди, если время погрузочных сократится с 50 до 40 мин.

Задача 30. Процесс изготовления досок на пилораме для хозяйственных нужд требует выполнения трех операций на определенных видах станков. Сначала производят распиливание бревен на доски, затем – обрезку доски, и в конце – обстругивание досок. Известно среднее время, через которое бревна поступают в систему на изготовление досок, – 30 ± 3 мин. Известно среднее время, затрачиваемое на выполнение каждой операции, и возможные отклонения от среднего времени, которые соответственно равны: 29 ± 1 , 20 ± 3 и 7 ± 3 мин. Среднее время для передачи материала от одного станка к другому и возможные отклонения от среднего времени составляют: от первого ко второму станку – 3 ± 2 , от второго к третьему –

4 ± 1 мин. Требуется определить основные характеристики системы:

- коэффициенты использования всех станков;
- среднее время пребывания изделия у каждого станка (канала);
- максимальное число изделий, ожидающих изготовления у каждого станка;
- средний размер очереди, то есть среднее число изделий, ожидающих обработки;
- общее число входов в очередь, т.е. общее число поступлений бревен на обработку за смену;
- среднее время пребывания изделий в очередях.

Оценить статистически существенно ли уменьшится длина очереди досок, если операция распила сократится с 30 до 20 мин.

ГЛАВА III. МЕТОД СИСТЕМНОЙ ДИНАМИКИ

3.1. Основные понятия метода системной динамики

Ученые давно пришли к выводу, что осознание и изучение элементов окружающего мира как системы наиболее адекватно действительности. Начал развиваться так называемый системный подход. Это наиболее сложный подход в науке, так как предполагает видение не просто отдельных обособленных объектов, а их изучение в объективной взаимосвязи и взаимозависимости с другими элементами рассматриваемой системы, а также ее взаимодействие с другими внешними системами, ее реакцию на внешние воздействия. Управление сложными системами в различных областях сопровождается определенными трудностями, так как это приходится делать в условиях непрерывного изменения состояния системы, в том числе и за счет реакции на наши управляющие воздействия. Помочь в управлении такими системами позволяет компьютерное моделирование. А метод, с помощью которого осуществляется моделирование, называется **методом системной динамики**.

Данный метод предложен в середине XX века американским ученым Джем Форрестером. В моделировании динамических систем метод системной динамики стал более широко применяться с развитием компьютерной техники, поэтому его часто называют **компьютерным моделированием**. Модели, создаваемые с помощью метода системной динамики, содействуют изучению сложной динамической системы, выявлять характер ее реакции на различные воздействия и предлагать наиболее эффективные решения.

Метод системной динамики по существу представляет собой совокупность различных методов, то есть он в определенном смысле реализует методологию системного анализа. Данный метод основан на применении методов математики, физики (при моделировании физических и технических систем), различных методах экономического анализа, анализа социальных систем и методы психологии различных субъектов (если моделируется экономическая система), метода нелинейной динамики, методов управления обратными связями, возникающими между элементами системы и др.

Системная динамика представляет собой совокупность принципов и методов анализа динамических управляемых систем с обратной связью и их применения для решения социально-экономических, производственных, организационных и технических задач. В системах поддержки принятия решений применение системной динамики позволяет объединить несколько функциональных пространств исследуемой системы в одно целое и обеспе-

чить организационный и количественный базис для выработки более эффективной управленческой политики.

Моделирование систем это непростой процесс, так как между элементами системы существуют обратные связи. Часто поведение системы непредсказуемо, то есть она развивается по непредвиденному сценарию. Например, с развитием офисной оргтехники не произошло уменьшение количества используемых бумажных носителей, как это ожидалось. Кроме того, моделирование осуществляют реальные люди, каждый из которых имеет свою логику мышления и по-своему видит логику функционирования системы, что также осложняет отображение реальной картины мира. Моделирование таких систем сложно, так как необходимо учесть всевозможные побочные реакции системы на управляющие воздействия. Так, например, преследуя цель накормить человечество, повысив урожайность культур за счет применения гербицидов, как побочный эффект получают загрязнение окружающей среды, нарушение биоценозов, и, как следствие, нанесение вреда здоровью человека.

То есть часто, воздействуя на систему, человек не может ясно себе представлять, как отреагирует на это система, так как причинно-следственные связи в ней носят не только прямой, но и обратный характер.

3.2. Событийно-ориентированный подход к моделированию систем

Моделируя систему, не всегда можно прийти к полному пониманию законов ее функционирования. Мы пытаемся это сделать путем установления причинно-следственных связей между элементами системы и событиями, происходящими в ней. Данный подход называется **событийно-ориентированным**. Таким образом, мы пытаемся найти как бы первопричину, определяющую результативные показатели работы системы, для того чтобы выработать управляющее на нее воздействие и достигнуть наилучшего требуемого нам результата. По разнице между наилучшим требуемым результатом и фактом мы формулируем проблему, которую собираемся решить с помощью моделирования системы. Так, например, в настоящее время доля сравнительно дешевой импортной свинины в рационе россиянина составляет 50 %, что ослабляет отечественного производителя свинины, ведет к сокращению ее производства, увеличению безработных в сельской местности, «вымиранию» деревень и т.д. Необходимо не только поставить проблему, но и выработать какое-либо управляющее решение. В данном случае оно может заключаться в том, чтобы, например, повысить импортные пошлины,

что приведет либо к сокращению импорта, либо к удорожанию импортной свинины. Тогда отечественные производители будут защищены и смогут более активно развиваться, насыщать рынок отечественной свининой, что со временем из-за увеличения предложения снизит ее цену. Однако данное управляющее воздействие со временем может привести к падению доходности данной отрасли из-за снижения цены и оттока производителей из отрасли. Таким образом, на позитивное управляющее воздействие со временем в системе может возникать ранее не прогнозируемая нами реакция. Это побудит нас снова вмешаться в ход событий и дать новое управляющее на систему, адекватное данному временному периоду, воздействие.

Моделируя систему часто трудно себе представить весь диапазон побочных эффектов, обратных связей и возникающих побочных эффектов. Особенно это касается таких сложных систем как социально-экономические, Они являются наиболее открытыми и находятся под воздействием разного типа закономерностей и законов – экономических, психологических, социальных и других. Непредсказуемые эффекты на управляющие воздействия возникают вследствие узости нашего представления о системе, о тех реакциях, которые в ней произойдут после нашего воздействия, ведь данным воздействием мы выводим систему из состояния баланса и она снова начинает стремиться к его обретению. Например, как это было показано выше, при повышении спроса на отечественную свинину из-за сокращения спроса на импорт, цена на нее сначала возрастет, что привлечет в отрасль дополнительных производителей, образуется некоторое временное состояние рыночного равновесия. А затем, вследствие неэластичности спроса при дальнейшем росте ее производства, цена начнет падать, и опять на рынке свинины наступит относительное равновесие из-за последующего сокращения ее предложения.

Устанавливая причинно-следственные связи в системе, мы часто думаем, что причина и следствие близки во времени, однако между различными причинами и следствиями существуют различные по длине временные лаги. Чтобы более адекватно смоделировать какую-либо систему, нужно хорошо себе представлять окружающую среду, в которой она действует, ведь одна, менее крупная система является как бы вложенной в более крупную. Чем шире охват нашим мышлением системы и ее окружающей среды, тем адекватней представляемая нами ее модель.

Метод системной динамики разработан для моделирования сложных экономических систем, которые представляются в виде моделей потокового типа. *Модели потокового типа* отражают взаимодействие элементов моделируемой системы с помощью потоков различной природы. Так, например, в такой социально-экономической системе, как регион государства можно вы-

делить потоки трудовых ресурсов и других видов населения, потоки материальные (например, перемещение материальных средств внутри и между предприятиями) Например, моделируя функционирование предприятия, можно выделить потоки сырья, денежных ресурсов, готовой продукции, трудовых ресурсов, информации и др. При взаимодействии различных потоков система непрерывно изменяется во времени, поэтому такую систему называют *непрерывной*.

3.3. Понятие и виды обратных связей в социально-экономических системах

Понятие обратной связи между элементами развито применительно к физическим системам. Однако об обратных связях можно говорить и по отношению к социально-экономическим системам, когда система реагирует на управляющие воздействия. Метод системной динамики, предложенный Д. Форрестером, является наиболее подходящим для реализации системного подхода в моделировании и в том числе, возникающих в системе в разные интервалы времени различные обратных связей. Можно сказать, что динамические процессы в системе и взаимодействие ее элементов во времени может быть выражено с помощью **петель обратных связей**.

Известны положительные и отрицательные петли обратной связи. **Положительные петли** обратной связи (Рисунок 29) предполагают взаимный рост (уменьшение) величин двух (или более) элементов системы. Например, с ростом прибыли предприятия, оно может увеличивать величину основных фондов, что через определенное время в свою очередь будет способствовать росту прибыли и так далее.

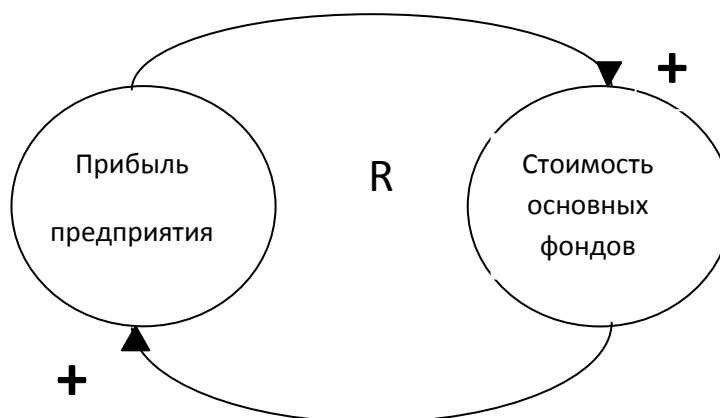


Рисунок 29 – Положительная петля обратной связи

Графически эту зависимость можно определить так (Рисунок 30).

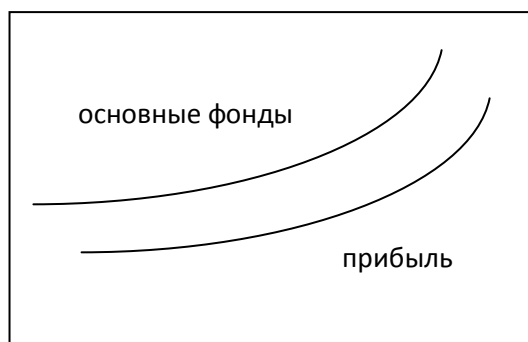


Рисунок 30 – Взаимозависимость между прибылью и основными фондами предприятия в динамике

Знак “+” говорит о том, что следствие положительно связано с причиной, ее увеличение ведет к увеличению следствия. Буква **R** - это обозначение полярности данного вида (self-reinforcing) – самоукрепляющая. Понятно, что в реальном мире такого не бывает. Значит существуют еще какие-то петли роста, включающие эти элементы. И это – отрицательные петли. Они сдерживают поступательное развитие событий в данном контексте и в конечном итоге определяют пределы роста.

Отрицательные петли обратной связи означает, что причина влияет на следствие с положительной динамикой, а обратная связь, реализующаяся через определенный временной лаг, является отрицательной. Например, рост цены карт, то есть величину его предложения. Но через определенное время будет прослеживаться обратная связь - влияние количества картофеля на рынке на его цену. И это будет отрицательная связь: чем больше предложение, тем ниже цена (Рисунки 31, 32).

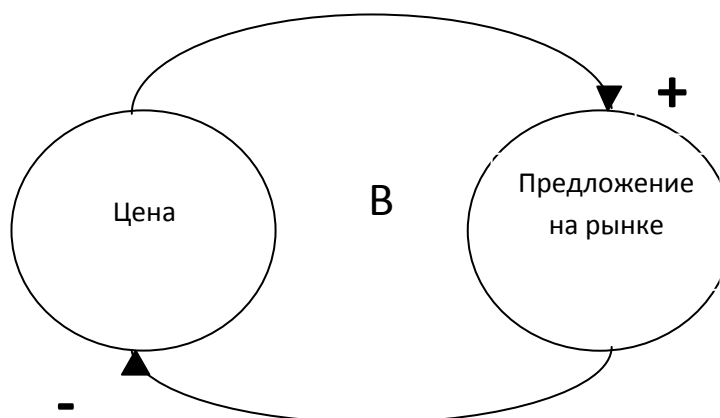


Рисунок 31 – Отрицательная петля обратной связи

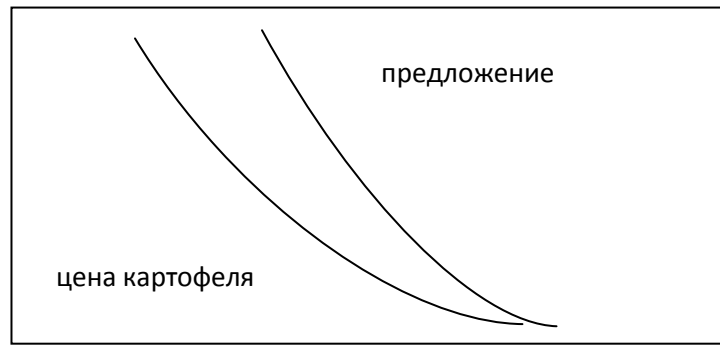


Рисунок 32 – Взаимозависимость между ценой и предложением картофеля

Обозначение **В** (balancing) означает отрицательную обратную связь. Опять же вследствие того, что данные элементы связаны не только этой петлей обратной связи, а какой-то (в данном анализе не отраженной) петлей или петлями положительной обратной связи, падение цены картофеля не будет продолжаться вечно.

Один и тот же элемент системы может участвовать сразу в нескольких положительных и отрицательных петлях обратной связи. Например, численность населения какого-либо региона на определенную дату является сначала аргументом, а потом функцией таких двух элементов как количество родившихся в данном году и количество умерших (задача несколько упрощена, так как в действительности на численность населения региона влияет количество эмигрантов и иммигрантов). Причем с количеством родившихся имеется положительная обратная связь, а с количеством умерших в данном году - отрицательная обратная связь (Рисунок 33).

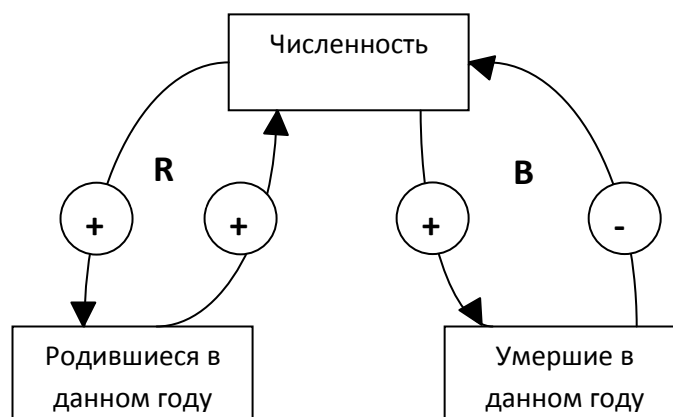


Рисунок 33 – Характер петель обратной связи с элементом системы

Символом R обозначена положительная обратная связь, а символом B - отрицательная. Так, с элементом системы «родившиеся в данном году» имеется положительная связь. Действительно, если сделать допущение, что темп рождаемости постоянен в течение нескольких лет, то увеличение численности населения приведет к увеличению абсолютной величины родившихся в каждом данном году. В свою очередь, увеличение количества новорожденных приведет к увеличению численности населения, что в свою очередь приведет к увеличению количества новорожденных в следующем году и так далее.

С элементом «численность умерших в данном году» имеется отрицательная обратная связь. При постоянной средней смертности (которая измеряется в количестве человек в расчете на одну тысячу населения), чем больше умерших, тем меньше численность населения, следовательно, тем будет меньше количество умерших в следующем году.

Моделируемая система может содержать тысячи петель обратных связей, как положительных, так и отрицательных. Они переплетены между собой. Обратное воздействие осуществляется через разные по протяженности временные лаги. И это еще раз обуславливает сложность системы. Поэтому моделировать поведение сложной системы в динамике можно только с помощью компьютерных средств. Это предполагает как можно более полное изучение обратных связей, приближение наших представлений о системе к реальному миру и к тем целям, которые мы предполагаем достигнуть в работе с этой системой – ее адекватное моделирование и управление ею.

Часто петли обратной связи имеют нелинейный характер, то есть поведение в динамике многих элементов системы является непредсказуемым.

Как указывалось выше, управляющее на систему воздействие производится для приведения фактического состояния системы в некое желаемое для нас состояние. В динамике, однако, реакция на управляющее воздействие не всегда приводит к желаемому результату. Поэтому последующие решения пересматриваются, становясь все более адекватными достижению желаемой цели, так как они пересматриваются исходя из реакции системы на предыдущие воздействия. Например, для преодоления кризиса, стимулируя рост спроса для роста производства, правительство государства решает дать государственный кредит под низкий процент предприятиям через банки (Рисунок 34). Однако обратная связь показывает, что банки направляют полученные от государства средства не в реальный сектор, а на зарубежные счета (то есть обратная связь показывает ошибку). Поэтому правительство корректирует пути достижения поставленной цели, стимулируя спрос за счет увеличения бюджетных трансфертов (пенсий, пособий и т.д.).

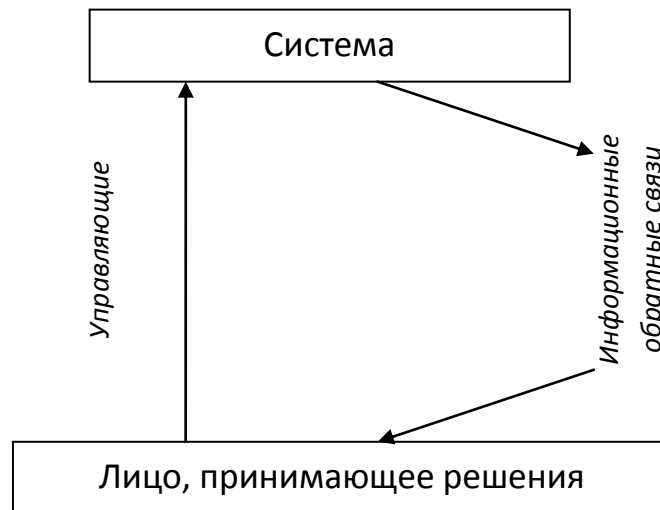


Рисунок 34 – Пример действия обратной связи.

На основе анализа характера обратного воздействия решения должны корректироваться. Однако это бывает не всегда. Это происходит, если претерпела изменения наше представление о характере причинно-следственных связей. То есть для того, чтобы управлять системой необходимо у лиц, принимающих решение какое-то представление о работе системы, окружающей ее среде, характере взаимодействия элементов системы и характере взаимодействия самой системы с окружающей средой. В научной литературе это представление называется **ментальной моделью**. Если ментальная модель в ходе получения нескольких обратных импульсов системы не меняется, то мы пытаемся достичь поставленных перед системой нами целей, управляя системой в рамках нашей старой ментальной модели. В моделировании системы и работе с ней по достижению определенной цели нельзя отрицать присутствия элементов психологии. В ментальной модели каждого человека существует элемент субъективизма, то есть личного, основанного на индивидуальном восприятии, количества информации, воспитании, умении анализировать ситуацию и т.д., восприятия мира, в то время как многие из людей (исследователей) полагают, что именно его ментальная модель объективно отражает реалии мира. Чем более широк кругозор ЛПР, тем быстрее обратные связи, поступающие от системы, меняют его ментальную модель. Тогда происходит изменение характера управляющих на систему воздействий.

Понятие ментальной модели - это одно из центральных понятий метода системной динамики. При создании моделей системной динамики одной из целей является создание как можно более адекватной реальному миру ментальной модели. Это является основой для более качественного принятия

решений по управлению системой. Однако в системной динамике пока не выработано универсального определения ментальной модели. Уместно привести определение ментальной модели, данное основоположниками метода системной динамики, Дж. Форрестером: «Умственное изображение или умственное описание на естественном языке, которое может формировать модель из общей организации и процессов.... Ментальная модель - это умственное изображение мира вокруг нас, которое мы несем в наших головах. Человек не имеет города или страны в своей голове. Он только выбрал понятия и связи, которые он использует, чтобы представить реальную систему. Ментальная модель размыта, незавершена, неточна».

Естественно, ментальная модель, существующая в сознании исследователя недостаточно учитывает динамику системы. Как указывал один из классиков системной динамики Стерман, ментальные модели недостаточно оценивают запаздывания между действием и реакцией системы, а также нелинейность причинно-следственных связей и характер их изменения в динамике. Они ограничены в представлении обратных связей, задержек во времени и процессов усиления. Поэтому реализуются эти модели с помощью компьютеров.

Ментальная модель является предварительным этапом разработки **концептуальной модели**, которая предполагает определенную формализацию ментальных представлений.

Системы с обратной связью имеют три основные характеристики: структуру, запаздывания и усиления. *Структура* представляет собой элементы системы и связи между ними. При моделировании подобных систем и описании взаимодействий элементов этих систем между собой и с внешней средой с помощью математического аппарата часто не учитываются *запаздывания*, то есть время, необходимое системе, чтобы отреагировать на новую поступившую информацию. Например, нужно время, чтобы рынок отреагировал снижением цены на повышение предложение какого-либо товара. В то же время запаздывания играют значительную роль в динамических характеристиках системы и при динамическом моделировании системы их нельзя не учитывать. *Усиления* в системе проявляются в том случае, если действие оказывается более сильным, чем это можно предполагать, исходя из ввода информации регулирующего решения.

3.4. Особенности моделирования сложных динамических систем

Изменяя состояние системы (воздействуя на нее) тем самым мы изменяем характер обратной связи. Если учесть, что в сложной системе таких пе-

тель множество и реакция системы на воздействия может быть как предсказуемой, так и непредсказуемой, а также нелинейность причинно-следственных связей, то можно себе представить, как сложно такую систему смоделировать, и что метод системной динамики, позволяющий это сделать, действительно является очень мощным инструментом в исследовании поведения систем и управления ими. Наибольшей сложностью характеризуются социально-экономические системы. Сложность системы определяется не только количеством входящих в нее элементов или иерархических уровней. Большую проблему в моделировании представляет собой отображение динамической сложности. Динамическая сложность – это полученный через определенное время результат взаимодействий в системе множества петель обратных связей, в которых обратные реакции происходят после управляющих импульсов через определенное время. Поэтому часто трудно проводить управляемые эксперименты с системой. Любая система развивается, и это тоже нельзя не учитывать при ее моделировании.

Представляет сложность проведение факторного анализа сложных систем, так как сложно провести эксперимент в чистом виде – с изменением факторного (анализируемого в данный момент) признака параллельно с разной скоростью и разной величиной запаздываний меняется множество других факторных и результативных показателей системы.

Поэтому после отслеживания реакции системы на управляющее воздействие необходимо снова и снова корректировать приемы воздействия на нее для достижения поставленной цели. После каждого «вмешательства» в систему она выходит из состояния равновесия (теряет устойчивость). Поэтому для того, чтобы выявить характер реакции системы на управляющее воздействие, за системой нужно наблюдать в течение определенного времени.

Обобщая вышеизложенное можно выделить несколько факторов, осложняющих моделирование и анализ сложных динамических систем:

- 1) разная скорость изменения в динамике различных элементов системы, характеризующихся определенными причинно-следственными связями;
- 2) обратные связи системы имеют различный характер после каждого нашего управляющего воздействия, что каждый раз требует его корректировки;
- 3) нелинейность, связанная как с логикой работы элементов системы (с увеличением количества рабочей силы на предприятии до определенного уровня прирост производства осуществляется быстрыми темпами, после данного уровня темп прироста производства начинает падать), а также взаимодействием многих факторов;

4) возможность выбора только определенных путей управляющего воздействия на систему, что исключает возможность проверки на системе результативности альтернативных управляющих воздействий. Например, применив тактику инвестирования в средства производства в ущерб инвестированию в рабочую силу, можно отследить характер реакции системы только на данное воздействие;

5) самоорганизация системы – так называемое «усиление» - неожиданно более мощная реакция системы на сравнительно незначительное по силе воздействие нее;

6) способность системы изменяться во времени, развиваясь как самопроизвольно, так и под воздействием управляющего воздействия. При этом одни элементы системы развиваются и становятся более значимыми для достижения конечного результата, другие же элементы становятся менее значимыми;

7) сложные причинно-следственные связи, распределенные во времени. Часто первой реакцией системы на управляющее воздействие бывает ухудшение ситуации, и только через определенное время – ее улучшение, согласно поставленной цели, и наоборот. Так, например, введение импортных пошлин на сельхозпродукцию в ближайшее время повысит цены на нее на внутреннем рынке (отрицательный для потребителя результат), но повышенные цены стимулируют производство внутри страны и через определенное время, когда рынок насытится продукцией, цена на нее может упасть.

8) сложность осмысления работы системы (создания ментальной модели).

Большинство социально-экономических объектов (предприятие, город, государство и т.д.) является сложными системами. Их можно представить в виде «черного ящика» - объекта с неясной структурой, судить о содержании которого можно по разнице входящих и выходящих импульсов. Проверять действие управляющих решений на реальном объекте – это длительный процесс, требующий больших финансовых вложений. Обратные связи, которые проверяются на объекте реального мира, очень медленны. Поэтому при изучении объектов реального мира создают **виртуальные объекты** с ясной исследовательской структурой и характером причинно-следственных связей. Виртуальные объекты позволяют «сжимать» время. На одном и том же объекте можно испытать действие разных управляющих воздействий. Рассматривая реакцию виртуальной системы в динамике, можно остановиться на какой-то итерации, чтобы скорректировать управляющее воздействие, то есть проводить управляемый эксперимент.

Виртуальный объект удобен, кроме всего прочего еще и тем, что на нем можно экспериментировать, моделируя чрезвычайные ситуации (техногенные катастрофы, крупные экономические кризисы и т.д.) для того, чтобы научиться выбирать наиболее подходящие решения в случае их возникновения в реальном мире.

Для построения виртуального мира используется формализация. Формализуя описание объекта реального мира вследствие его сложности можно использовать знания различных специалистов, так как имеющиеся в нем связи многообразны, и, как правило, находятся в области множества различных областей знаний (наук). Поэтому здесь как бы соединяются знания и ментальные модели множества специалистов, и создается довольно полная, хорошо и ясно структурированная модель реального объекта. Получаемая модель значительно превосходит умственные способности одного человека. Разница в ментальной и формализованной модели приводит к совершенствованию их обеих.

Система с обратной связью предполагает, что она будет реагировать на изменение внешней среды или какое-то управленческое решение. В свою очередь реакция системы будет влиять на внешнюю среду или на дальнейшие решения. И такой процесс является возобновляющимся и непрерывным (вспомним рыночный механизм).

3.5. Этапы имитационного моделирования методом системной динамики

Поскольку реальные системы существуют не только в пространстве, но и во времени, динамическое моделирование наиболее адекватно отражает процессы, происходящие в реальной жизни. *Динамическое моделирование* системы представляет собой изучение деятельности объекта моделирования как информационной системы с обратной связью. Если моделируется производственная система – предприятие, то его динамическая модель отражает взаимодействие элементов его организационной структуры, влияние управленческих решений и времени запаздывания в решениях и действиях на результат. В модели отражается взаимодействие товарных, денежных, информационных потоков, потоков средств производства и рабочей силы.

Динамическое моделирование является основой для выработки наиболее эффективных управленческих решений. Аналогично прочим подходам динамическое моделирование включает следующие этапы:

- 1) определение задачи;
- 2) выделение факторов, которые определяют цель функционирования системы;

- 3) выявление общей логики функционирования системы, характера взаимосвязи между ее элементами, установление обратных связей;
- 4) построение математической модели, включающей источники информации и взаимодействие элементов системы;
- 5) компьютерный «прогон» модели;
- 6) выявление адекватности составленной модели реальной системе;
- 7) корректировка модели с целью ее большего соответствия системе;
- 8) проверку на модели результатов возможных управленческих решений с целью выработки наиболее успешной стратегии развития;
- 9) совершенствование функционирования реальной системы на основе результатов моделирования;

Методология динамического моделирования основывается на:

- теории управления информационной системой с обратной связью;
- исследовании процессов принятия решений;
- экспериментальном моделировании сложных систем;
- использовании компьютера как средства имитации работы реальных систем.

3.6. Основные элементы модели системной динамики

Так как функционирование системы моделируется с учетом фактора времени, модели потокового типа относятся к *динамическим* моделям. Для динамического моделирования характерен экспериментальный подход. Данный подход применяется вследствие слабости математического аппарата для нахождения решений в сложных ситуациях. Модель дает описание того, каким образом существующие условия, в которых в данный момент находится система, приводят через определенное время к другим условиям. Для выявления работы системы на модели проводят серию экспериментов.

Динамика системы моделирования описывается в виде *потоковых сетей* (финансов, сырья, информации и др.). Потоковые сети состоят из уровней, соединенных взаимосвязанными потоками, которые, протекая по всей системе, изменяют значения уровней. При моделировании работы предприятия уровнями будут являться товары на складе или в пути, количество работников на предприятии, производственные площади и др. Уровни описывают величины, непрерывные по диапазону своих значений и дискретные во времени. Они представляют собой те значения переменных в данный момент времени, которые они имеют в результате накоплений в системе из-за разности между входящими и выходящими потоками. Они присутствуют во всех шести потоках, на которые условно можно разделить потоки предприятия:

материалов, заказов, денежных средств, рабочей силы, оборудования и информации.

Скорость изменения потока называется *темпом*. Темп показывает как изменяется уровень за интервал времени, равный шагу моделирования, т.е. темпы характеризуют динамику моделируемой системы. Темпы отражают активность процесса, а уровни отражают состояние, которое определяется активностью в системе.

В методе системной динамики закон изменения темпа задается функциональной зависимостью от аргументов (уровней), значения которых на данный момент времени известны. Эта зависимость называется *функцией решений*. Например, производительность труда может быть выражена через количество выпущенных товаров (уровень) и время, затраченное на их производство (константа). Функция решения может иметь сложный вид. Так, например, решение о найме рабочей силы может быть принято на основе следующих уровней: количество рабочих на предприятии, средний темп поступления заказов, количество невыполненных заказов, наличие оборудования на предприятии и др.

Отличительной чертой уровня от темпа является то, что если систему рассматривать в статическом режиме, уровни, в отличие от темпов, все равно останутся. Например, при прекращении поступления и отправки товаров со склада, склад как таковой останется, и в нем сохранится какой-то запас продукции. Аналогично можно сказать и о таких уровнях как рабочая сила, денежные средства и др. Не все относительные величины, измеряемые в расчете на единицу времени, являются темпами. Так, например, такой уровень как «средняя производительность труда за прошлый месяц», измеряемая количеством продукции в единицу времени и отражающая эффективность организации производства на предприятии, является именно уровнем, а не темпом. Если представить себе, что деятельность по производству продукции остановится, то это никак не отразится на средней производительности труда на данном предприятии за прошлый месяц.

Значения уровней определяются заново для каждого из последующих интервалов решений. Предполагается также, что между моментами времени, для которых решаются уравнения, уровни изменяются с постоянной скоростью, но их значения на этом промежутке времени не вычисляются.

Потоки (уровни, темпы и вспомогательные переменные) задаются следующими уравнениями:

$$X_{t_i + h} = X_{t_i} + h \cdot V(t),$$

где t – модельное время; h – шаг моделирования; X_{t_i} , $X_{t_i + h}$ – значение уровня в моменты времени t_i и t_{i+h} ; $V(t)$ – скорость изменения уровня (величина уровня в единицу времени).

Например, уравнение уровня «текущий запас товаров на складе» может быть записано так:

$$Z_t = Z_n + \Delta T (Po - B),$$

где Z_t – запас товара текущий; Z_n – запас товара на некий начальный момент времени; ΔT – интервал времени; Po – поставки товара на склад за время ΔT (темп входящего потока); B – выбытие товара со склада за время ΔT (темп выходящего потока).

То есть запас товара на текущую дату равен начальному запасу плюс произведение разности между темпами входящего потока и выходящего потока на интервал времени ΔT . Темпы потока измеряются числом единиц за какой-либо интервал времени. При этом шаг интервала времени одинаков для всех потоков.

Уравнения уровней не зависят друг от друга. Каждое из уравнений зависит только от состояния уровня на предыдущий момент времени и темпов потока в течение интервала времени ΔT .

Перейдем к *уравнениям темпов* (функциям решений). По типу решений уравнения темпов могут относиться к *явным* или *неявным*. *Явными решениями* называют сознательные решения людей по вопросам процесса управления системой. Например, какое количество рабочих нанять, или какое количество сырья закупить. *Неявные решения* зависят от состояния системы, а не от конкретных решений. Например, уровень производства продукции определяется такими факторами как численность рабочих, обеспеченность их средствами производства, количеством поставок материалов и др. Неявным решениям предшествуют явные. Знание различий между ними помогает выявить важные этапы в потоках информации и выработке стратегий. Разница в структуре этих уравнений отсутствует.

Уравнения темпов регулируют действия, которые должны произойти в системе за следующий интервал времени. То есть определяется действие, которое будет управлять темпом потока в течение предстоящего интервала времени. Уравнения темпов зависят только от значений уровней в начальный момент времени.

Взаимодействие в системе происходит при последующем воздействии темпов на уровни, которые затем в свою очередь оказывают влияние на темпы в более поздние интервалы времени. В качестве примера уравнения темпа можно привести следующее:

$$Tn = \frac{Km}{Bn},$$

где Tn – темп выходящего потока (единицы в неделю); Km – требуемое покупателю количество товара, находящегося в данный момент времени в процессе производства (единицы); Bn – средняя продолжительность времени производства и поставки данного количества товара покупателю (недели).

Данное уравнение показывает, какое значение будет иметь темп на протяжении определенного следующего интервала времени.

На принятие управленческих решений влияют определенные факторы. От того, насколько точно они учтены и определены в модели, зависит ее адекватность. Поэтому важно установить, какие факторы наиболее значительны. Далее для каждого из факторов определяют направление и значимость его влияния. Как выделить наиболее существенные факторы?

Одним из способов является наблюдение за работой модели при наличии фактора и без него. При этом нужно учитывать, что даже если фактор не напрямую влияет на решение, может существовать обратная связь, характеризующая влияние решения на фактор. И это действие может усугублять влияние данного фактора на дальнейшие изменения решения. Например, при задержках в поставке товаров покупатели начинают покупать их впрок, т. е. увеличивают спрос. В результате этого еще больше увеличивается отставание выполнения заказов на поставку данного товара.

Один и тот же фактор может оказать в различные периоды времени как отрицательные, так и положительные последствия на результат. Так, например, в долгосрочном периоде рост цен на продукцию определенного вида стимулирует рост ее производства. В то же время в краткосрочном периоде это может вызвать временное сокращение производства вследствие увеличившихся инвестиций и неосвоенности инвестиционных проектов.

Функции решений могут быть представлены как в линейном, так и в нелинейном виде. Для моделирования функционирования предприятия наиболее адекватными бывают нелинейные зависимости. Часто это – нелинейные функции второго порядка:

$$R = aX^2 + b(Y)$$

Например, зависимость роста темпа производства на предприятии от количества рабочих является параболической, так как темп роста производства сначала увеличивается с ростом количества нанятых рабочих, а затем (после преодоления точки оптимального соотношения рабочих и техники) темп роста производства начинает снижаться.

Первая причина нелинейности в том, что влияние фактора, воздействующего на решение, не просто пропорционально этому фактору. Вторая причина – в том, что решение зависит не порознь от двух или большего числа вводимых переменных, а является результатом произведения той или иной взаимозависимости этих переменных.

Практически невозможно сразу составить адекватную системе модель. Потребуется ее отладка. Как оценить адекватность модели системе?

При прогоне первоначально составленной модели ее неадекватность системе может проявиться в таких характеристиках как, например, количество рабочих может оказаться отрицательной величиной; некоторые потоки могут изменить направление и двигаться противоположно логике; значения переменных могут находиться за логически допустимыми в реальной жизни пределами.

Другим способом проверки адекватности является введение в модель заведомо ложных условий, не соответствующих реальной системе, и проведение прогона модели с этими значениями.

Виды уравнений в моделях системной динамик

Переменные, используемые при структуризации функциональных зависимостей выходов от уровней и входов, называют *дополнительными* (*вспомогательными*) переменными. Они вводятся в модель тогда, когда уравнение темпа может быть очень сложным и его разбивают на отдельные части. С их помощью можно определить в отдельности многие факторы, учитываемые при выработке решения. Вспомогательные уравнения подставляются одно в другое и потом – в уравнения темпов. Вспомогательные уравнения решаются на начальный момент времени **после** решения уравнения уровня, но **до** решения уравнения темпа. Вспомогательные уравнения решаются, в отличие от уравнений уровней и темпов, в строгой последовательности.

Дополнительные уравнения отражают переменные, не являющиеся частью структуры модели, но используемые при печати или графическом изображении необходимых исследователю величин.

Уравнения начальных условий используются для определения исходных значений всех уровней и некоторых темпов. Они должны быть решены до начала первого цикла решения уравнений. Данные уравнения используются для вычисления значений одних констант в начальный момент времени, исходя из значений других констант, и они решаются один раз перед началом каждого прогона модели.

Экзогенные переменные

Производственные и социальные системы тесно взаимодействуют с внешней средой. Поэтому модели, отражающие работу такой системы, называются *открытыми*. На их работу оказывает влияние внешняя среда, которая отражается в модели в виде *экзогенных* переменных. Это такие переменные, значения которых не зависят от внутренних параметров системы. Вопрос состоит в том, какие переменные должны рассматриваться как экзогенные. Например, при моделировании работы предприятия А экзогенными переменными не являются заказы на продукцию, сделанные предприятием В, так как деятельность предприятия А в большой степени зависит от этих заказов. Поэтому это следует рассматривать как внутренние переменные.

При построении эконометрических моделей в них иногда включается несколько величин, взятых из реальной жизни. Такие величины называют «экзогенными вводами» и рассматривают их как переменные, которые используют в уравнениях модели, но они не являются величинами, полученными с помощью этих уравнений. Их можно использовать как своего рода проверочные сигналы, дающие возможность узнать, как реагирует модель на определенное поведение окружающей среды. Сами по себе они не зависят от того, что происходит в модели, и **их не используют для того, чтобы сделать модель наиболее приближенной к реальности**. Они используются для того, чтобы установить реакцию изучаемой системы на предположительные изменения во внешней среде. Более одной экзогенной переменной применяется редко.

Проведение моделирования методом системной динамики включает следующие этапы:

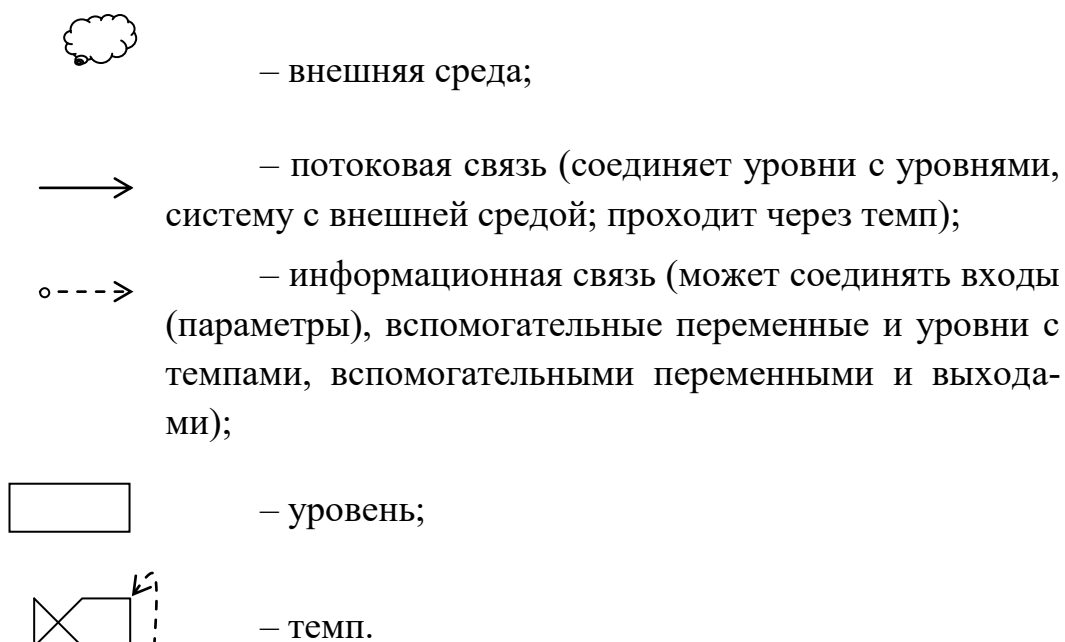
1. Задание начальных условия модели (значения уровней в начальный момент времени, начальный момент времени), шага моделирования и длины интервала моделирования.
2. На данный момент модельного (системного) времени t рассчитываются значения всех темпов и вспомогательных переменных. (Это делается автоматически, с помощью выбранной исследователем системы моделирования).

3. Системное время увеличивается на шаг моделирования h .
4. Рассчитываются значения всех уровней на данный момент системного времени.
5. Пункты 1–4 выполняются до тех пор, пока не пройдет весь интервал моделирования.

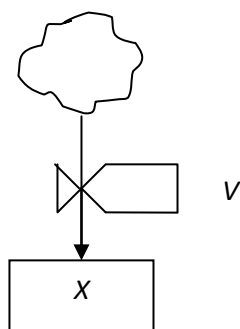
Построению модели должно предшествовать установление причинно-следственных связей факторов, включенных в модель, которые предварительно должны быть структурированы путем введения в модель вспомогательных переменных.

В методе системной динамики используются *специальные диаграммы причинно-следственных связей*, образующих модель. Диаграммы причинно-следственных связей строят следующим образом. На основе логического анализа системы выделяют переменные модели. Затем проводят их попарную классификацию, устанавливая «причину» и «следствие», которые потом соединяют стрелками. Над стрелкой ставят знак минуса или плюса («минус» ставится, если рост причины ведет к уменьшению следствия, «плюс» – если рост причины ведет к росту следствия). Так могут быть выделены уровни с обратной связью.

В методе системной динамики существует общепринятая техника графического описания моделируемых систем:



Например, потоковая сеть с узлом (X), потоковой дугой и темпом (V) может выглядеть так:



Ниже элементы модели системной динамики будут рассмотрены на конкретных примерах.

3.7. Основное меню системы Vensim

Основное меню программы Vensim включает следующие элементы:

File Edit View Layout Model Options Windows Help

Меню **File (Файл)** и **Edit (Правка)** позволяют осуществлять стандартный набор операций, предлагаемый соответствующими опциями других программ (Рисунок 35).

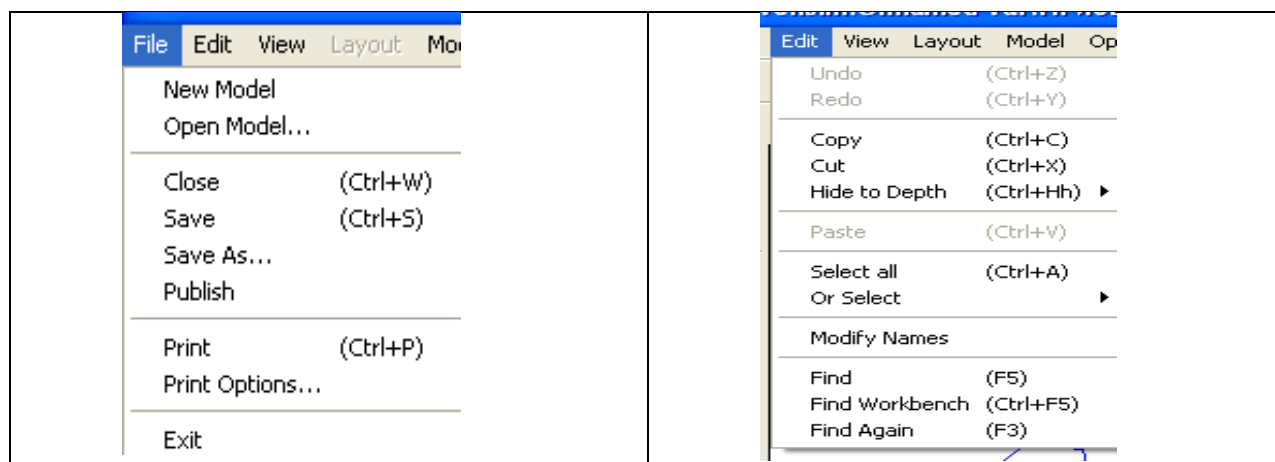


Рисунок 35 – Меню File (Файл) и Edit (Правка)

Остановимся подробнее на опциях меню Правка. Меню Edit. Для Vensim PLE и Vensim PLE Plus меню Edit включает следующие функции (Рисунок 36).

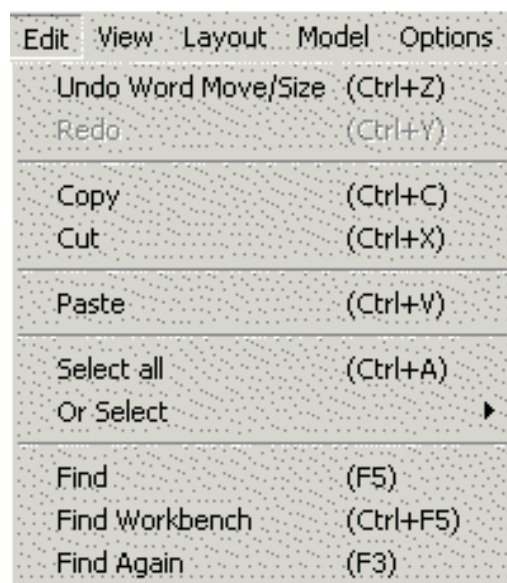


Рисунок 36 – Меню *Edit*.

Для того, чтобы были активны пункты данного меню, например, копирование, предварительно нужно выделить в модели определенный элемент.

Undo (отмена/отмена последней операции)

Команды Undo и Redo позволяют отменять изменения, которые недавно произведены.

Copy (копировать)

Копирует выбранные слова (или элементы структуры модели) в буфер обмена. Скопирована может быть также часть модели, выделенная предварительно. Скопированная информация может быть вставлена либо в другие модели, либо в текстовые процессоры и др.

Cut (вырезать)

Копирует выбранные слова в буфер обмена и одновременно удаляет их. При использовании этой команды открывается окно с вопросами относительно того, что именно нужно сделать (Рисунок 37).

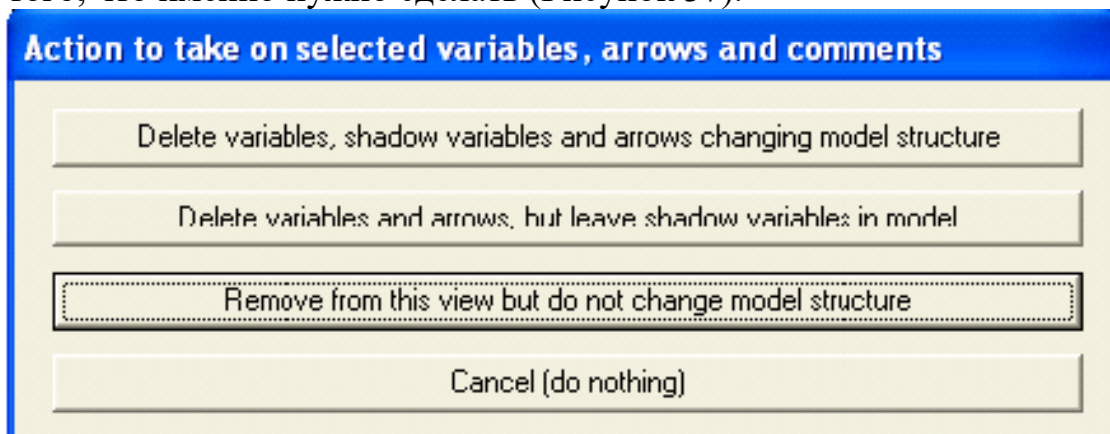


Рисунок 37 – Варианты действий с выбранными элементами модели

Действие, заданное по умолчанию Remove from this view but do not change model structure, удаляет выделенное, но не изменяет структуру модели. Выделенный фрагмент будет удален. Можно удалить также стрелки (связи) отдельно. Если у элемента модели есть входящие и исходящие стрелки (связи), то данный элемент делается теневой переменной, но не удаляется. Эта опция не доступна для Vensim PLE или PLE Плюс.

Кнопка Delete variables, shadow variables and arrows changing model structure (удаление переменных, теневых переменных и стрелок, изменяя структуру модели), удаляет выбранный элемент модели.

Кнопка Delete variables and arrows, but leave shadow variables in model (удаление переменных и стрелок, исключая теневые переменные) удалят все выбранные обычные переменные и все выбранные стрелки от модели, кроме теневых переменных.

Выбор клавиши **Del** удаляет элемент, не помещая его в буфер обмена.

Paste (вставить) производит вставку скопированного ранее в буфер обмена.

Select All (выделить все) выделяет все объекты модели. Команда **Or Select** предлагает выпадающее меню того, что нужно выбрать (Рисунок 38).



Рисунок 38 – Выпадающее меню команды Or Select.

Find (Найти) используется для поиска элемента модели (как видимого, так и скрытого).

Find Workbench (Найти Инструментальные средства) выполняет аналогичную операцию, но ищет только элемент модели, с которым недавно работали.

Find Again (Найдите Снова) повторяет нахождение переменной, выбранной на предыдущем этапе командой Find.

Меню View

Опции меню представлены на Рисунке 39.

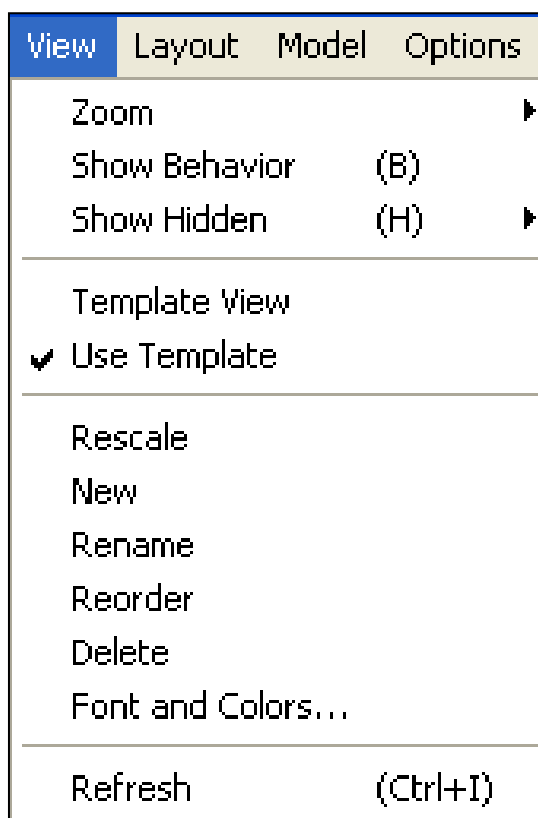


Рисунок 39 – Опции меню View

Zoom (Масштабирование) позволяет изменять масштаб отражения схемы модели (нужно выбрать процент увеличения из списка или указать свой процент увеличения).

Show Behavior (показ поведения элемента модели) – отменяет или назначает (если поставлена «галочка») показ на эскизе модели графика изменения элементов модели.

Rescale (Перемасштабирование) позволяет менять масштаб представления модели по осям X и Y, устанавливая процент в диалоге (Рисунок 40).

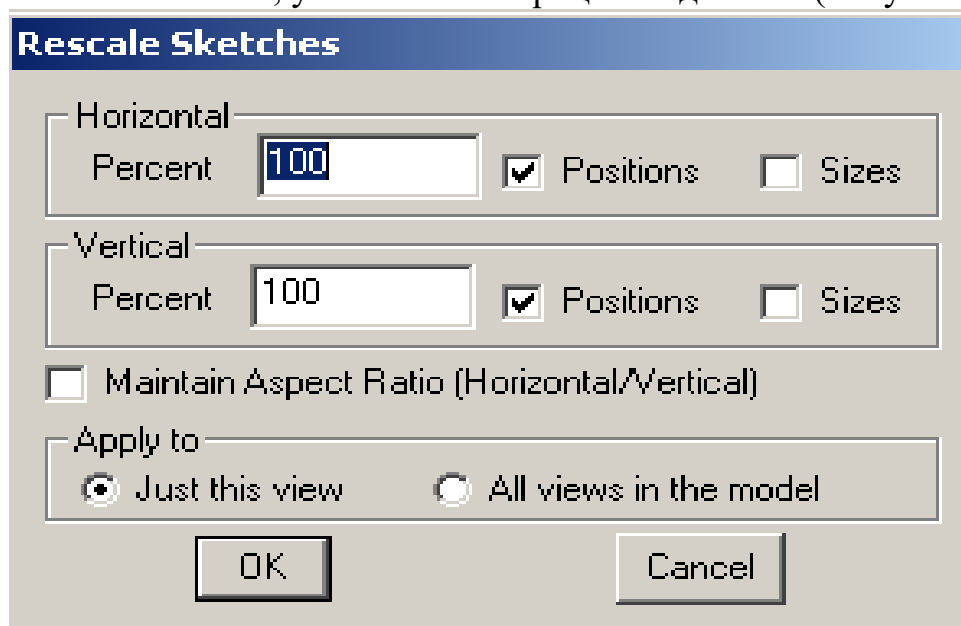


Рисунок 40 – Диалог Rescale

«Галочка» в **Positions (Положения)**, заставляет положения слов и стрелок изменяться. «Галочка» в **Sizes** будет менять размеры слов (названий элементов модели.). Галочка в **Maintain Aspect Ratio** приводит в соответствие горизонтальный размер изображения модели вертикальному. «Галочка» в **Just this view** позволяет применять данные настройки изображения модели к данному окну или ко всем окнам модели.

New (Новый) создает новое окно (представление), которое будет добавлено к основанию списка представлений. Новому представлению модели будет присвоен номер (Рисунок 41).

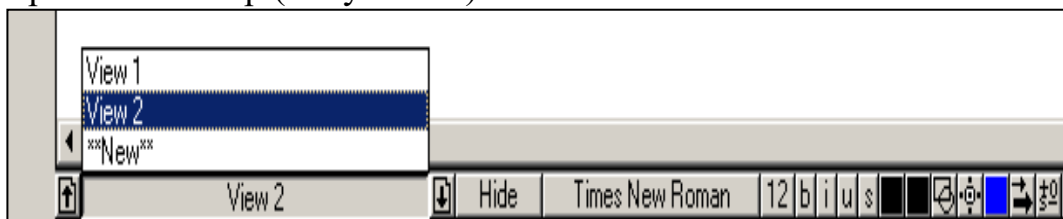


Рисунок 41 – Присвоение номера новому окну.

Rescale (Переименовать) позволяет переименовывать текущее окно. Окнам можно присвоить не цифровые, а символьные имена, с максимальной длиной 32 символов.

Reorder (Перезаказ) позволяет повторно заказывать окна модели. Это делается с помощью диалога **Перезаказа Списка**.

Delete (Удалить) удалит данное окно без возможности его возврата.

Font and colors...(Шрифт и Цвета) открывает следующее окно (Рисунок 42).

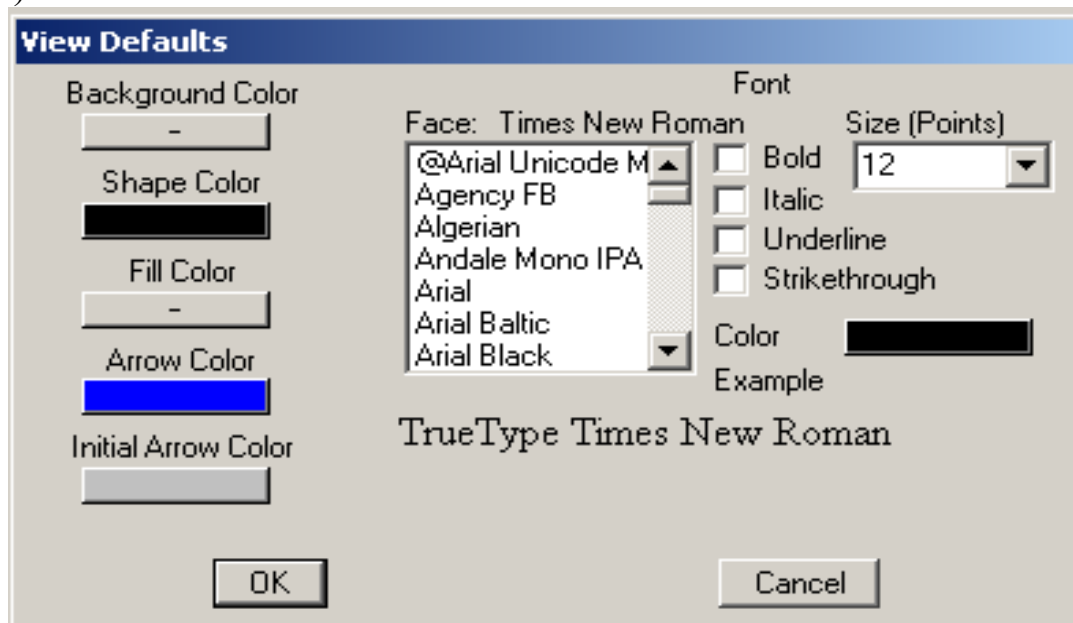


Рисунок 42 – Установление шрифта и цвета.

Окно имеет набор опций, позволяющий выбирать из раскрывающейся палитры различные цвета для изображения стрелок в модели, уровней (Fill Color), темпов (Shape color), начальных переменных (Initial Arrow Color),

стрелок (Arrow Color), а также общий фон, на котором изображена модель (Background Color).

Refresh удаляет некоторые элементы на эскизе модели, которые не являются логически частью эскиза модели.

Меню Model (Модель)

Для версий PLE Плюс PLE меню **Model** представлено на Рисунке 43.

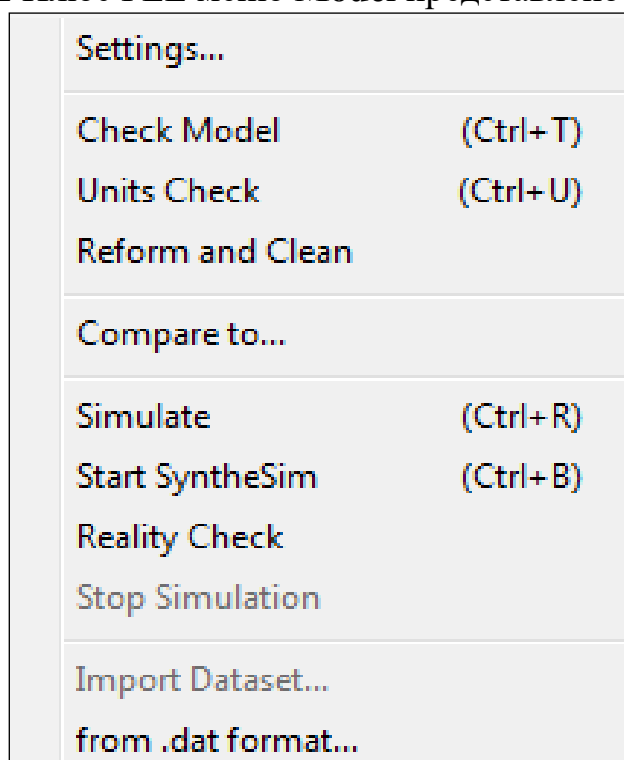


Рисунок 43 – Опции меню Model

Setting (Параметры настройки) позволяют открывать диалог Параметров настройки и устанавливать такие параметры как начальное время моделирования, конечное время, единицу модельного времени и шаг (Рисунок 44).

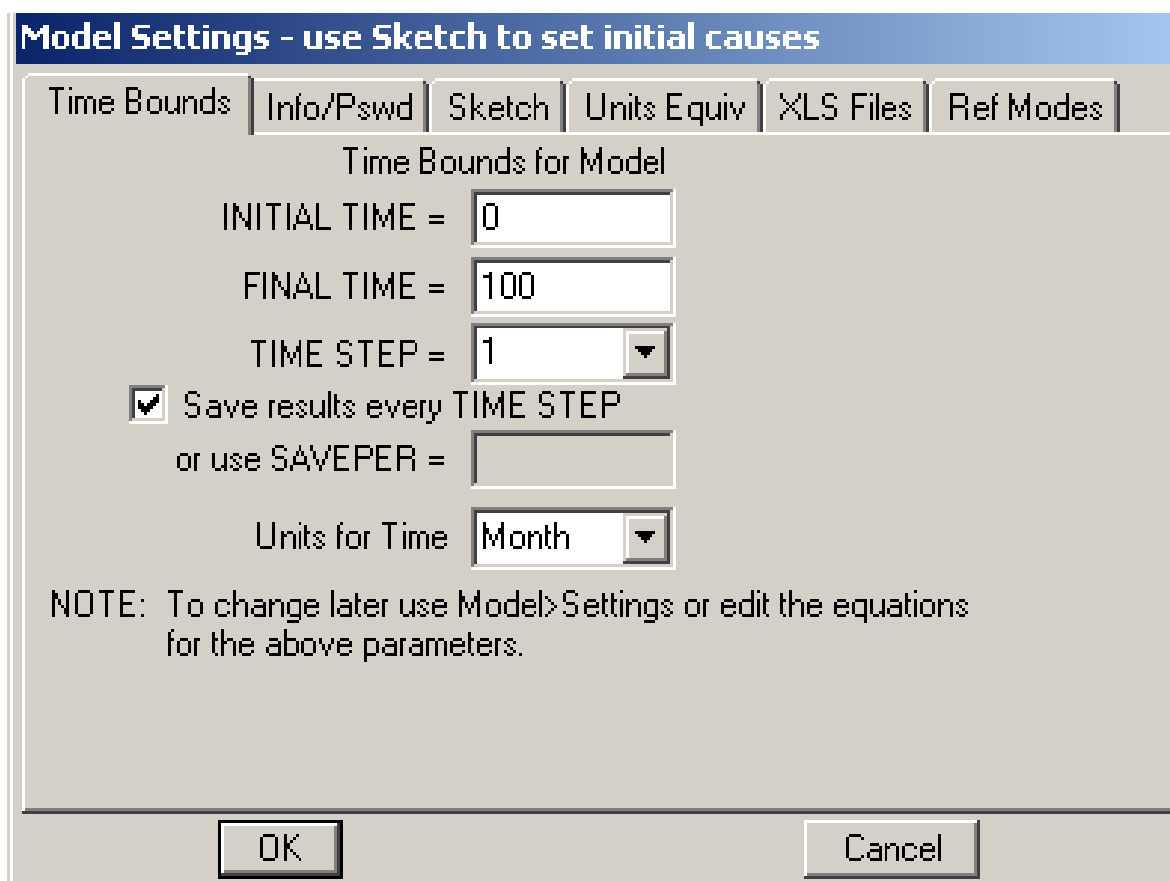


Рисунок 44 – Установление горизонта и шага моделирования


Окно также содержит предложение сохранять (отображать) результаты прогона модели каждый шаг времени.

Check Model (проверка модели) проверяет модель. Если в ней имеются ошибки, выдается сообщение о характере ошибки.


Units Check (проверка единиц измерения) проверяет логическое соответствие единиц измерения и выдает сообщение о характере ошибок в отображении единиц измерения.

Compare to (Сравните С ...) позволяет сравнить данную модель с другой моделью. При этом запрашивается открытие какой-либо другой модели.

Simulate (Моделировать)

В версиях *Vensim – Standart, Professional и DSS* эта опция открывает диалог контроля за моделированием и прогоном модели (аналогично кнопке на панели инструментов ). В версиях *Vensim PLE и PLE Плюс* эта опция открывает диалог только для прогона модели.

Опция **Start SyntheSim** также служит для осуществления прогона мо-

дели, но в режиме SyntheSim (аналогично нажатию кнопки ). Опция **Run Game (запуск игры)** производит поргон модели в режиме игры. Опция **Sensitivity** производит прогон модели с оценкой чувствительности к

какому-либо фактору, **Reality Check (Проверка в реальных условиях)** открывают диалог проверки модели в условиях прогона в реальных условиях.



Опция **Stop Simulation** (остановить прогон) аналогична кнопке .

Меню Options

Опции данного меню представлены на Рисунке 45.

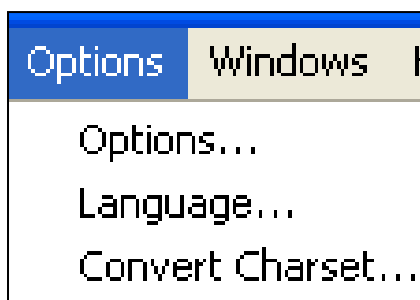


Рисунок 45 – Опции меню Options.

Данное меню позволяет следующее. **Options (Функции)** открывает диалоговое окно **Global Options**. Это диалоговое окно с закладками позволяет устанавливать различные опции для настройки Vensim. **Language (Язык)** открывает диалог, который позволяет выбрать другой язык для пользовательского интерфейса программы Vensim.

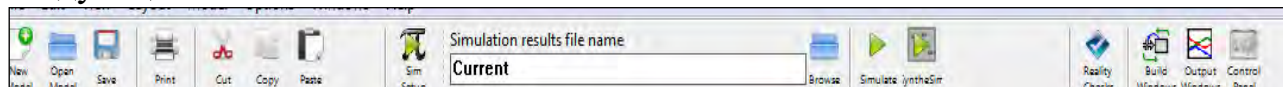
Compare to (Сравнить с...) позволяет сравнить текущую модель с другой моделью. Диалог открывает окно открытия какой-либо, ранее созданной модели. После выбора данной модели открывается окно сравнения (сравниваются названия переменных, горизонты и шаг моделирования, используемые функции и другие параметры модели).

from .dat format позволяет импортировать данные в модель из файлов с расширением *dat* (см. ниже).

3.8. Панель управления файлом модели, прогонами и настройками модели

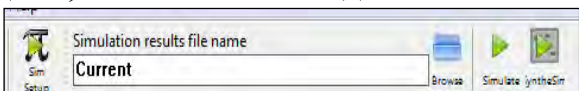

3.8.1. Управление файлами и панель настройки

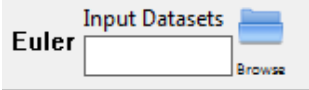
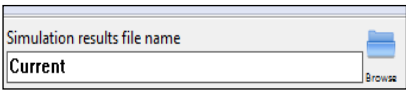


Управление файлами, в которые помещены модели, осуществляется следующей после основного меню панелью:




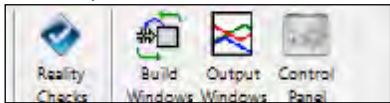
По функциональности данную панель можно разделить на три группы кнопок. Первая – традиционные опции для работы с файлами – *Создать новую модель, Открыть модель, Сохранить файл, Распечатать, Удалить, Ко-*

тировать, Вставить. Последние две (Копировать и Вставить) могут относиться как ко всему файлу модели, так и к части модели.


Вторая часть панели –  – инструменты управления прогонами модели. Кнопка  предполагает использование инструмента импорта данных из файла. При этом на данной панели слева запрашивается путь к файлу, из которого будут импортированы дан-

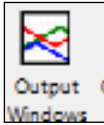
ные: . Поле Current –  используется для сохранения прогона модели под определенным именем (можно провести несколько прогонов модели с измененными данными и каждый из них должен называться по-разному). Рядом расположена кнопка  (*Browse*) для указания пути, где будет сохранен данный прогон. Кнопка  (*Simulate*)

позволяет осуществить прогон модели. Кнопка  также позволяет провести прогон модели, но с визуализацией на элементах модели шкал управления константами и графиками динамики изменения значений элементов модели (то есть в режиме Synthesim).

Третья часть панели  содержит различные опции настройки и тестирования модели. Первая кнопка открывает новую функцию программы (появилась только в версии 6) – оценка адекватности модели. Ранее можно было проводить оценку адекватности без использования специального аппарата только эмпирически.

Следующие три кнопки открывают три вида окон:

1. **Build Windows**  используется для создания моделей в Vensim, когда модель может быть представлена в виде нескольких сопряженных блоков, каждый из которых открывается в отдельном окне. Даная опция пока доступна только в коммерческих версиях программы – *Vensim Professional* и *Vensim DSS*.

2. **Output Windows**  создается с помощью инструментов *Vensim's Analysis*, и отображает окна графиков, таблиц и списков (Рисунок 46). Output Windows появляются после нажатия кнопки на панели Analysis tool. Она позволяет отображать сразу нескольких таких окон, открывающихся

после нажатия соответствующей кнопки на вертикальной панели инструментов. Одновременно могут быть открыты десятки окон.

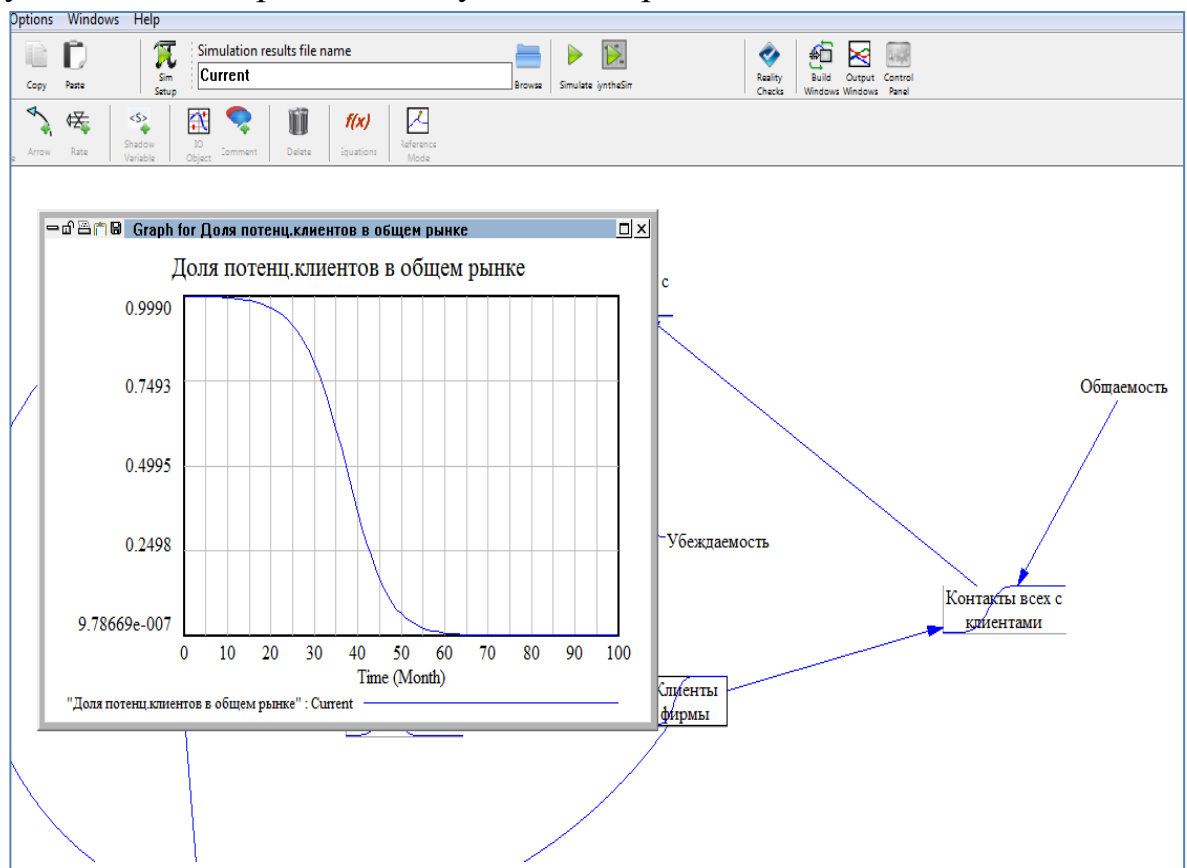



Рисунок 46 – Окно графика

При этом окна графиков открываются обычным образом – с помощью кнопки . Несколько окон можно открыть, не закрывая ранее открытые.

Output Windows позволяет просмотреть все открытые с помощью аналитической панели окна одновременно (Рисунок 47).

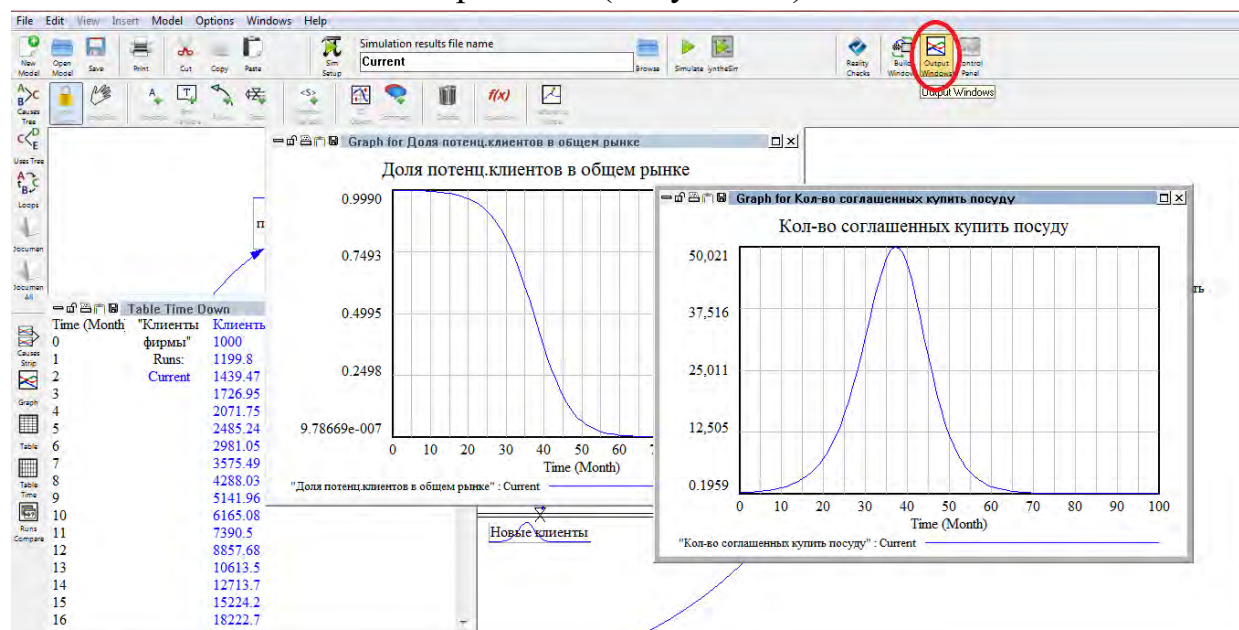





Рисунок 48 – Несколько одновременно открытых окон

Определенное окно можно вывести на первый план с помощью комбинации кнопок: Ctrl+Shift+Tab. Переключение между окнами осуществляется кнопками: Ctrl+Tab.

3. **Control Windows**  включает панель управления – *Control Panel*, вкладка диалогового окна используется для управления внутренними настройками *Vensim*, и *Subscript Control*, используемые для определения и выбора индексов в *Vensim Professional* и *DSS*.

Панель управления позволяет управлять выходной информацией, которая открывается с помощью аналитической (вертикальной) панели (*Инструменты анализа*). Открыть Пульс управления можно также кнопкой на панели инструментов  (в версиях программы до шестой – эта кнопка выглядит так: ). При ее нажатии открывается следующее окно (Рисунок 49).

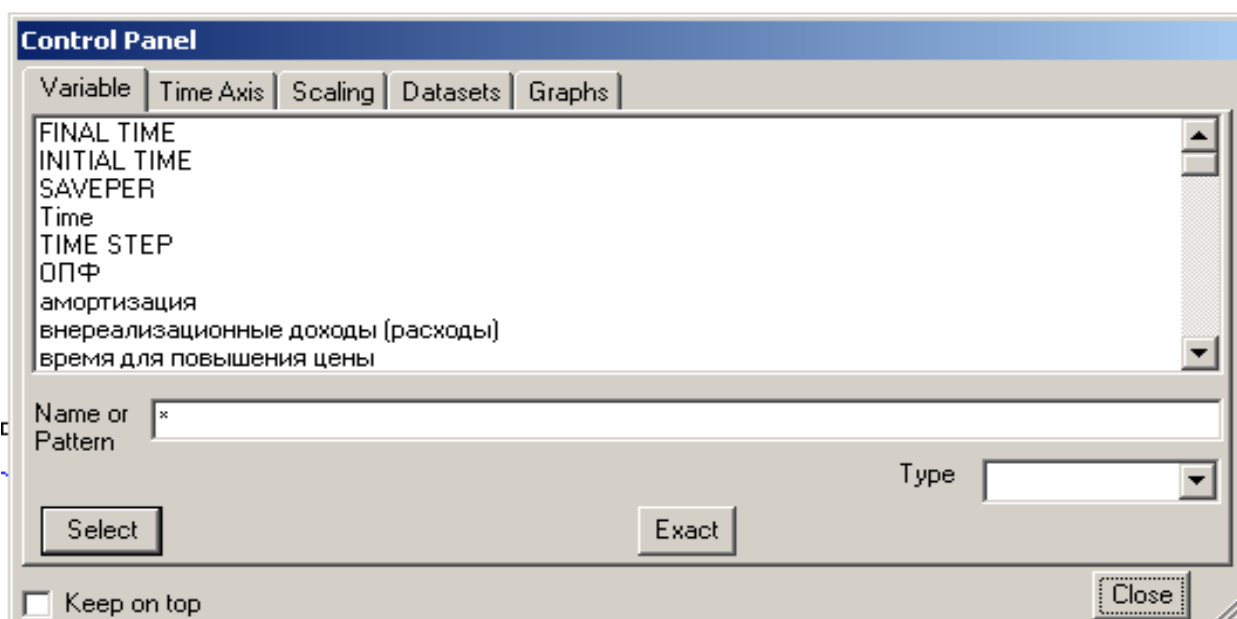


Рисунок 49 – Вкладки панели настройки модели

Вкладка **Переменная (Variable)** позволяет выбрать из списка тот элемент, с которым нужно производить какие-либо действия. Для выбора нужно дважды щелкнуть на ее имя или набрать имя в поле **Name or Pattern** и нажать клавишу **Exact**, если элемент указан точно. Можно также набрать несколько первых букв его названия и добавить звездочку. *Type* позволяет выбрать тип переменной модели (*All*, *Lavel*, *Auxiliary* и др.). При выборе типа переменной только выбранный вид переменных покажется в списке (Рисунок 50).

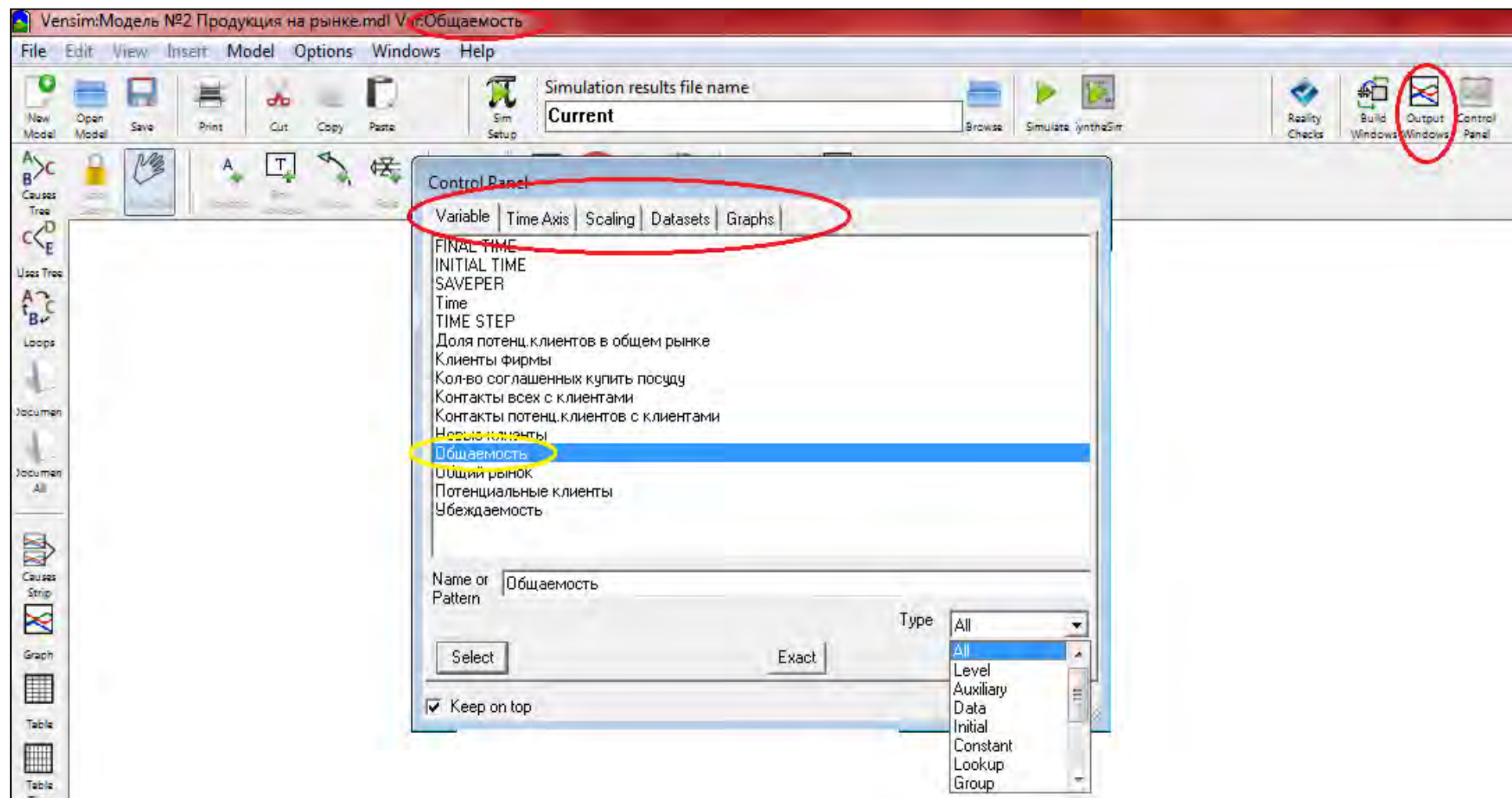


Рисунок 50 – Список выбранных переменных модели

На вкладке представлен список элементов модели. Нажатием на имя нужной переменной она выбирается в рабочую область для отображения графика или таблицы. Рабочая область переменной отображается в верхней части окна Vensim. Можно сортировать элементы по типу (*Type*), что позволяет увидеть все элементы (All) или элементы с определенным параметром, например уровень (Level), чтобы искать все переменные, содержащие «*Level*».

Используя вкладку **Ось Времени (Time Axis)** можно изменить масштаб изображения в установленном в модели диапазоне времени, удерживая нажатой клавишу *Shift*, и двигаясь мышью через весь график. Также с помощью *Time Axis* можно управлять диапазоном времени (Рисунок 46).

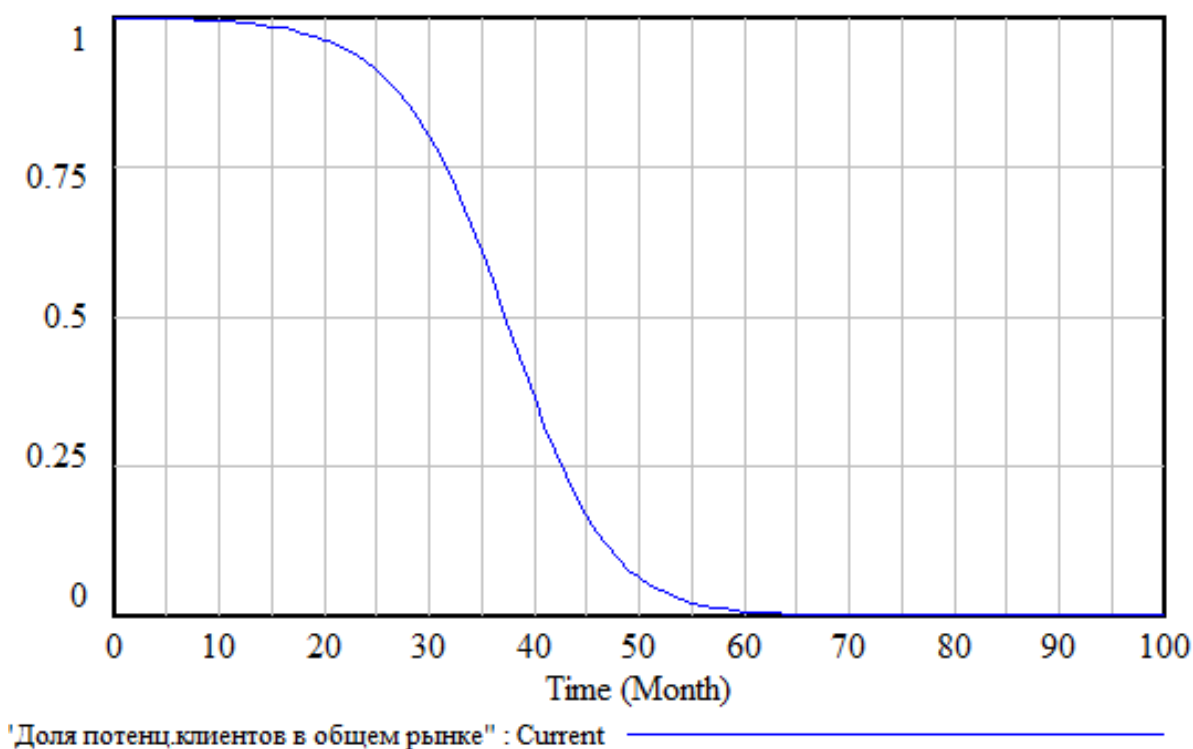


Рисунок 46 – Динамика показателя «Доля потенциальных клиентов в общем рынке»

С помощью вкладки **Time Axis** можно также устанавливать начальное, специальное и конечное время (Рисунок 47).

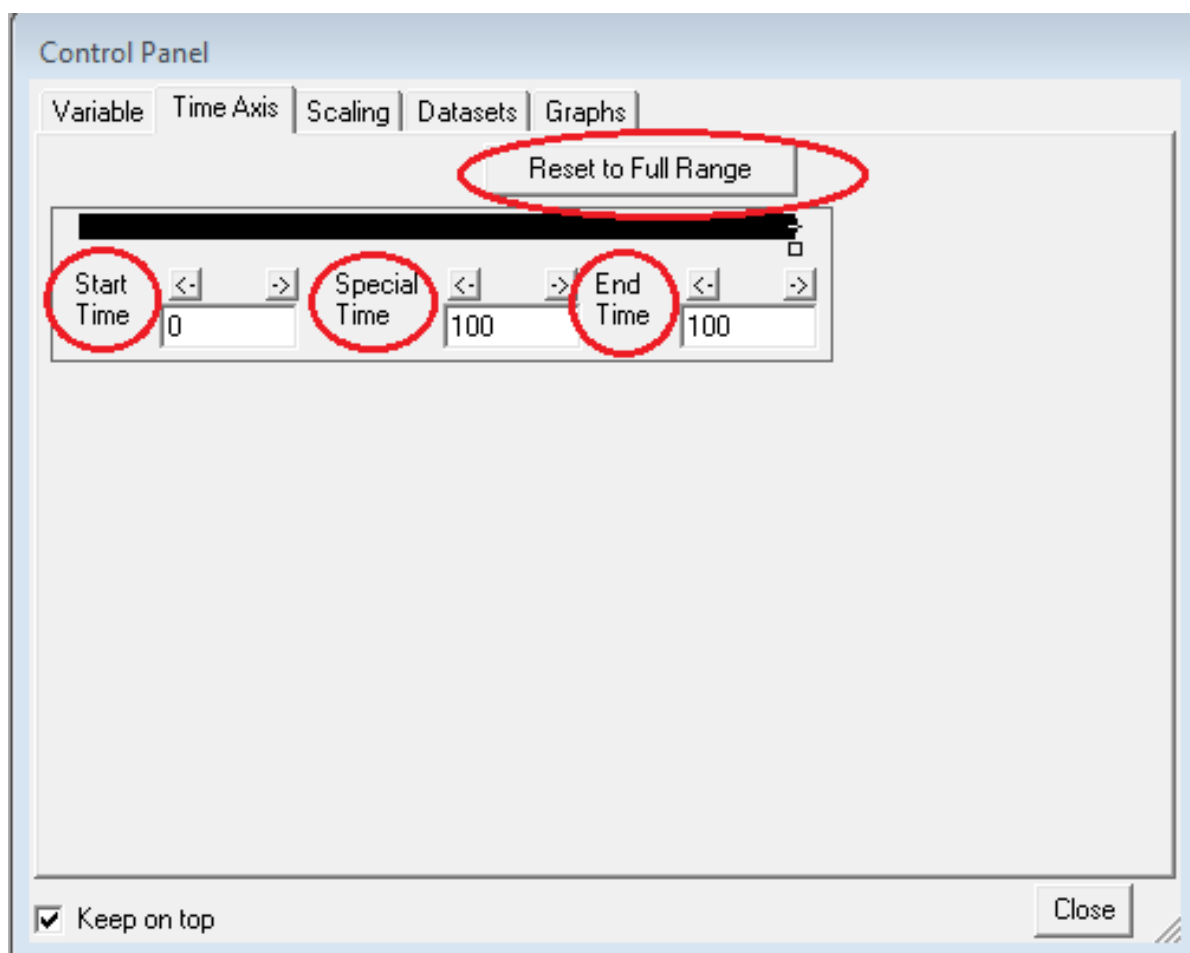
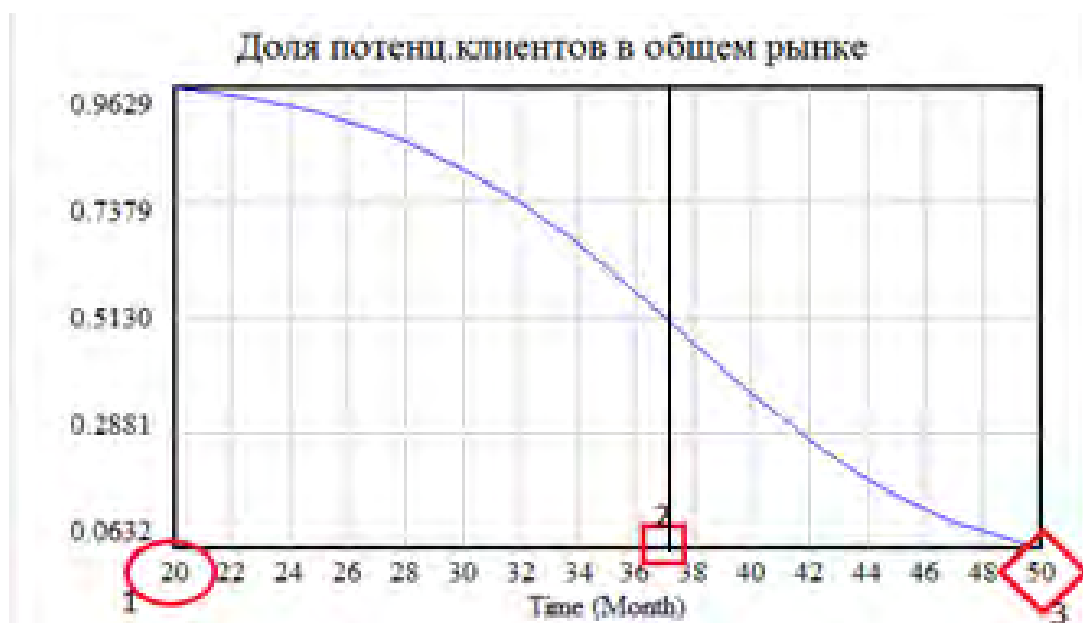


Рисунок 47 – Установка начального и конечного времени

Для изменения временных границ открывается вкладка *Time Axis* и устанавливается необходимый диапазон времени (Рисунок 48). Показатель *Special Time* устанавливает конкретную точку из диапазона, на графике это отображается вертикальной линией.



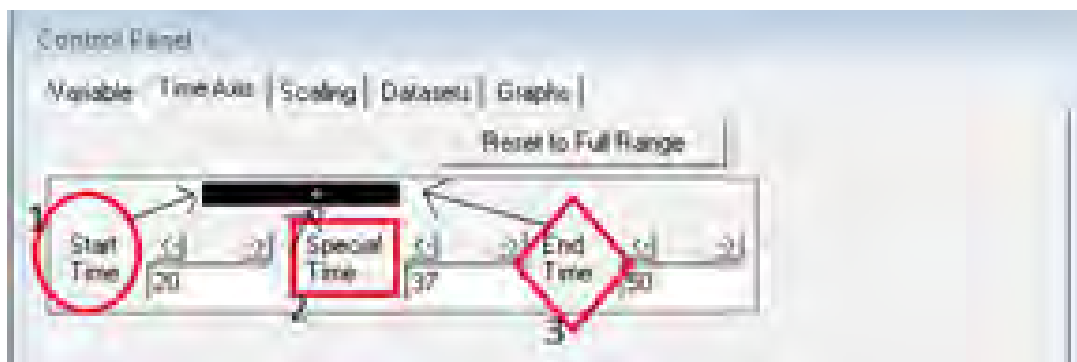


Рисунок 48 – Графическое изображение установленного диапазона времени

Поле *Horizontal Divisions* задает количество горизонтальных делений (Рисунок 49). Данное поле используется по отношению к *Strip*, *Sensitivity* и *Bar* графикам. Здесь можно задать номер или нажать на стрелку «вниз» и выбрать из появившегося списка. При этом изменение количества делений не изменит графики, которые уже созданы, а будет распространяться только на новые графики.

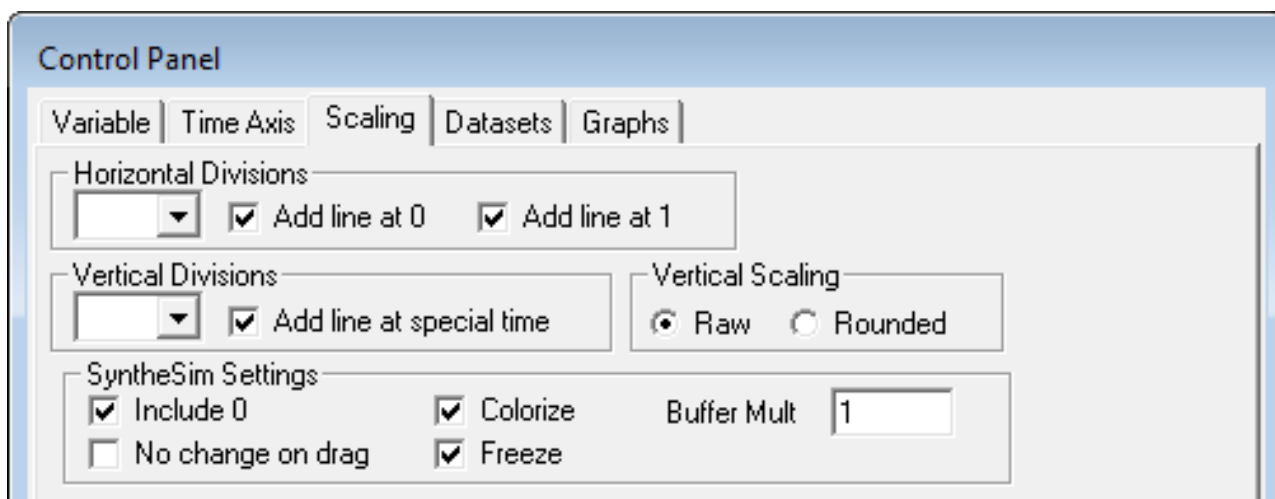


Рисунок 49 – Формирование вида осей графика.

По умолчанию количество делений задается в зависимости от оси времени. Можно задать количество делений сразу для всех графиков или выбрать «1» для каждого графика отдельно.

В поле *Add line at 0* при поставленной галочке добавляется горизонтальная линия графика в Y-значение 1,0. Это применимо только для графиков, которые поднимаются от отрицательных значений к положительным. Если установлено *Add line at 1*, добавляется горизонтальная линия графика в Y-значение 1,0 (Рисунок 50).

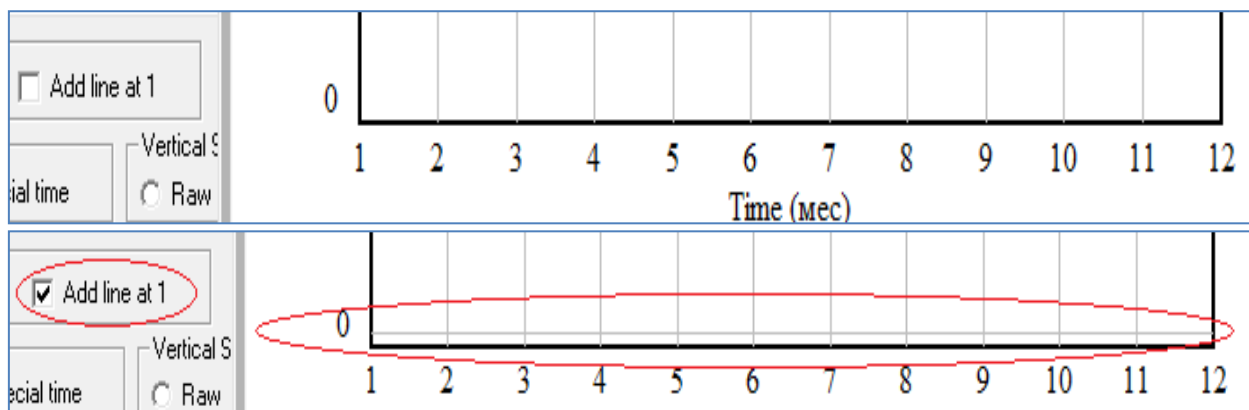


Рисунок 50 – График с добавленной горизонтальной линией

Vertical Scaling определяет количество вертикальных делений. Это изменит все графики, за исключением пользовательских графиков, для которых было задано количество вертикальных делений. При установке *Add line at special time* добавляется вертикальная линия по месту нахождения специального времени. *Vertical Scaling (Raw или Rounded)* определяет внешний вид вертикальной шкалы:

Raw – верхнее значение на шкале является максимальным значением переменной, нижний минимум (Рисунок 51).

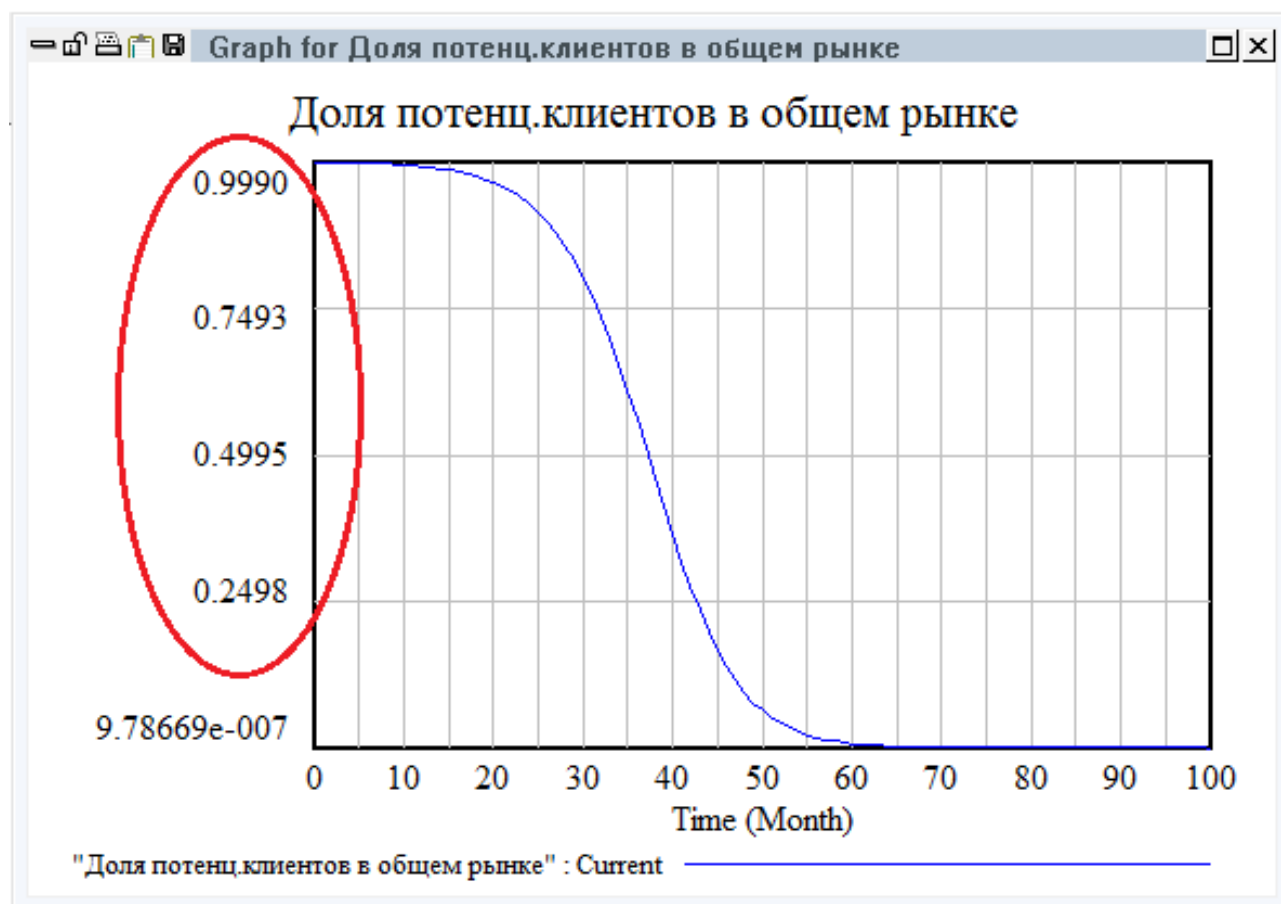


Рисунок 51 – График переменной, начинающийся с ее максимального значения

- *Rounded* – цифры округляются для удобства чтения (Рисунок 52).

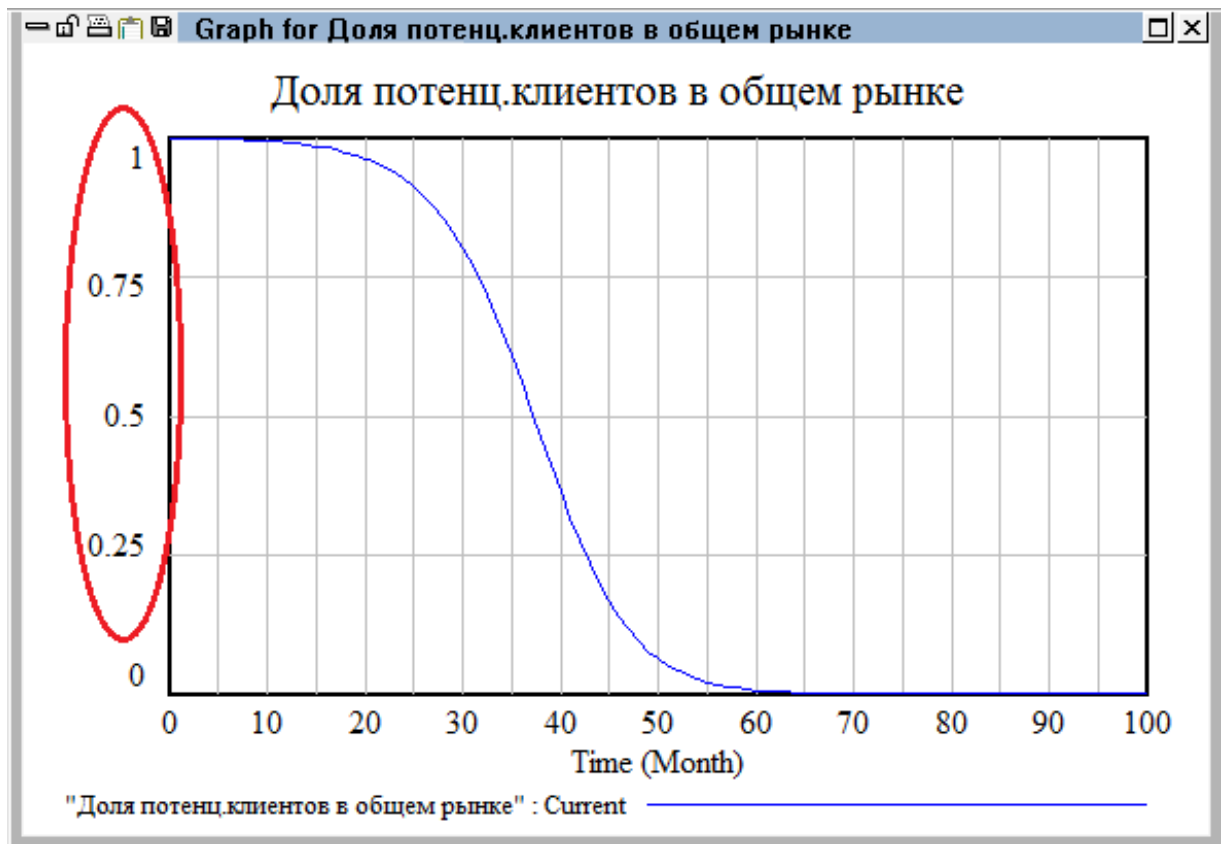



Рисунок 52 – Округленные значения оси ординат.

«Галочка» в поле *Include 0* будет способствовать тому, чтобы график всегда начинался с нуля, независимо от значений переменных. При выборе *Colorize*, область графика переменной при прогоне модели в режиме

SyntheSim (с помощью кнопки ) будет мигать красным, показывая увеличение размера шкалы, и синим цветом – при уменьшении масштаба шкалы (Рисунок 53).

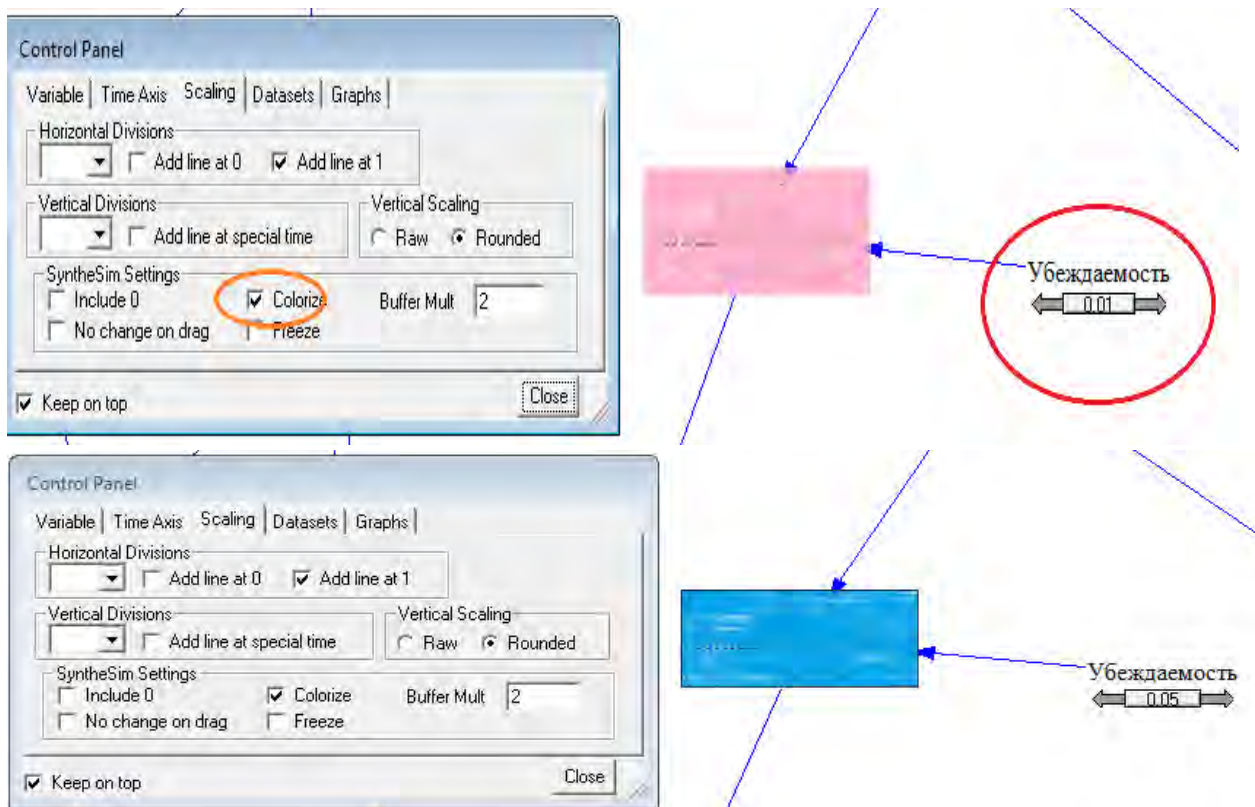


Рисунок 53 – Колорирование переменной при ее отрицательной или положительной динамике

При выборе *No change on drag* графики остаются неизменными, пока не будет отпущена кнопка мыши, после чего они будут отображены (Рисунок 10). При выборе *Freeze* миниатюрные изображения графиков при прогоне в режиме **SyntheSim** будут сохранять первоначальные масштабы. Чтобы масштабировать их, нужно перейти на другой вид и вернуть или загрузить и выгрузить набор данных из *Control Panel* (Рисунок 54).

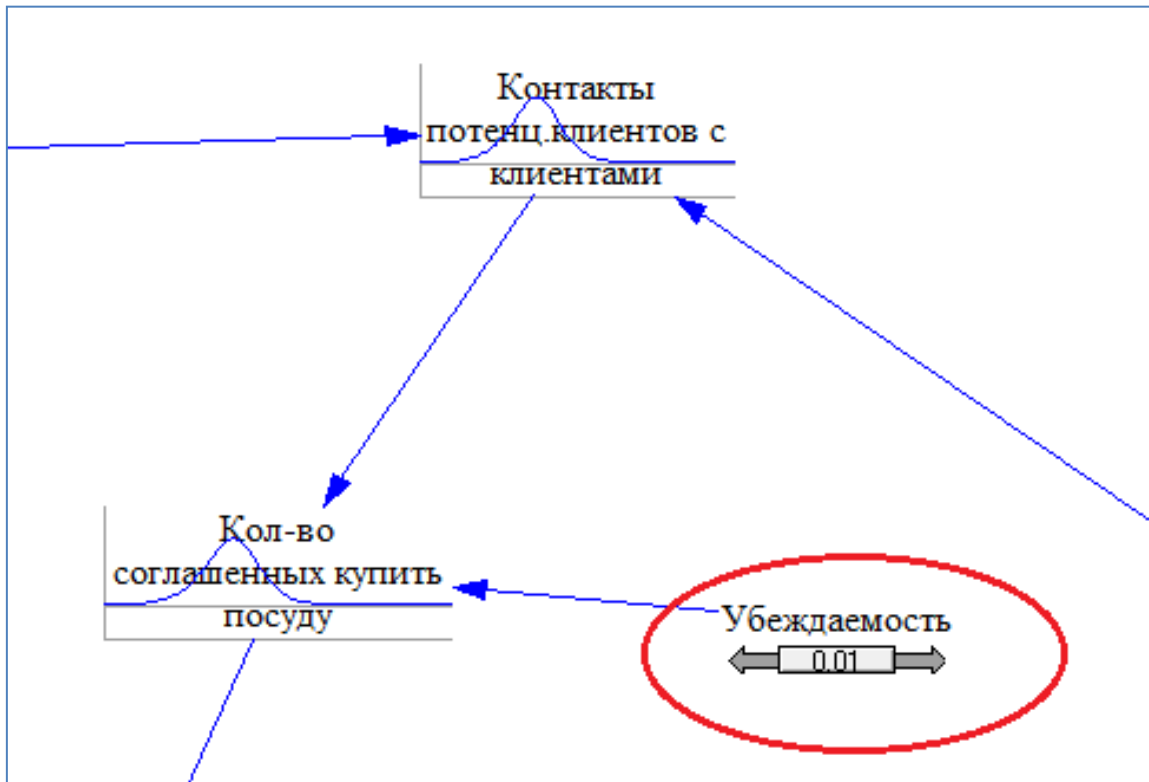


Рисунок 54 – Масштабирование элемента модели при прогоне в режиме SyntheSim.

• *Buffer Mult* определяет сколько дополнительного места нужно отвести для графика определенного элемента модели в случае его прогнозируемого роста в будущем (Рисунки 55, 56).

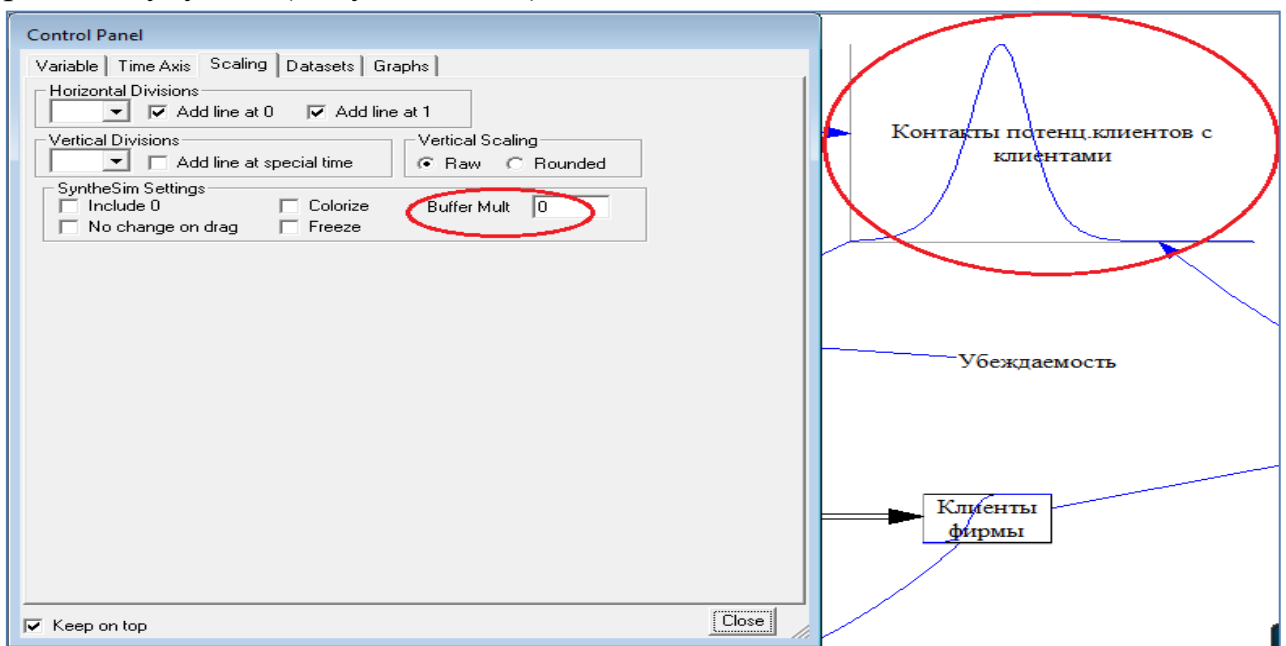


Рисунок 55 – Форматирование изображение графика.

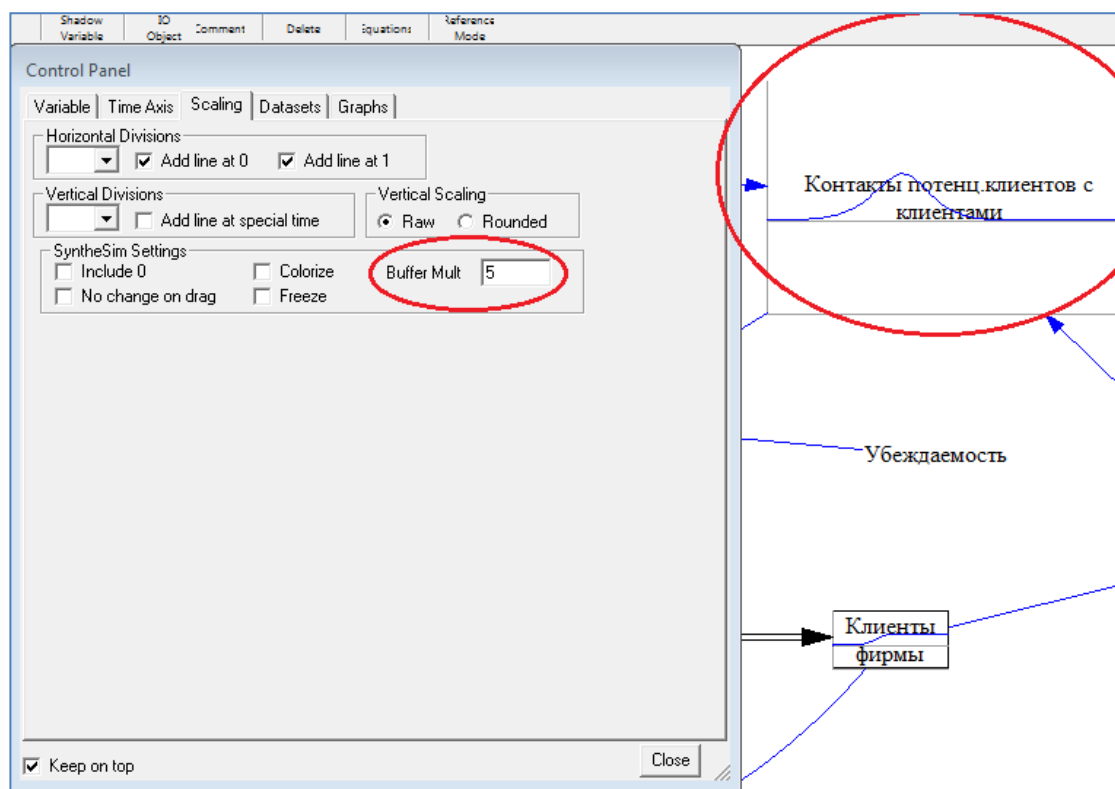


Рисунок 56 – Настройки форматирования

Вкладка Наборы данных (Datasets)

При пользовании инструментами *Анализа*, например, просмотре динамики изменения какой-либо переменной данная опция позволяет просматривать результаты тех прогонов модели, которые в данный момент необходимы пользователю. Например, на вышерасположенном рисунке показано, что на графике будут выдаваться результаты двух прогонов – прогона с названием «наценка 2,5 руб» и прогона с названием «наценка 10 руб». В окне Available – Info показаны все доступные прогоны, а в окне Loaded – Info – те прогоны, которые выбраны (Рисунок 57).

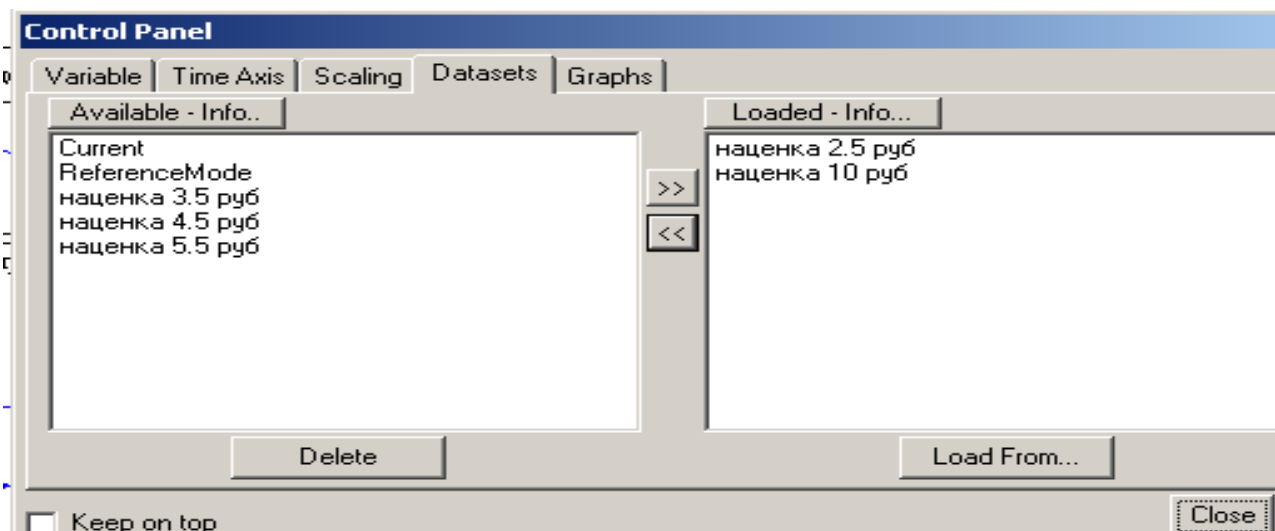


Рисунок 57 – Возможности вкладки набора данных

Здесь также можно удалить набор данных, полученных при прогоне с помощью кнопки **Удаление (Delete)**. То же самое можно сделать, удаляя файлы прогонов из той папки, где они были сохранены, но это – более трудоемкая операция.

Кнопка **Загрузка из... (Load From)** позволяет загружать в график набор данных из какого-либо файла, путь к которому нужно указать.

3.8.2. Графики и таблицы пользователя

В программе Vensim можно отображать графики необходимых пользователю переменных с помощью опции **Граф пользователя (Custom Graphs)** на одной координатной плоскости. Графы пользователя открываются с помощью специальной функции **Управление графом (Graph Control)** на панели управления (**Control Panel**) опции Windows панели инструментов (Рисунок 58).

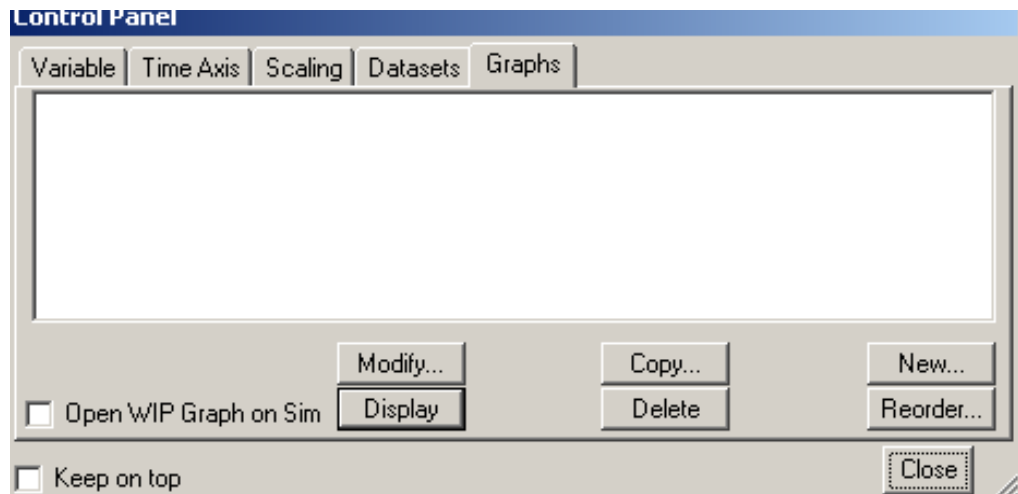


Рисунок 58 – Вкладка Graph

Граф пользователя создается с помощью следующих шагов:

Провести прогон модели, дав ему название, например, *Прогон1*.

- Выбрать вкладку **Graphs** в окне Панели управления;
- Нажать кнопку **New**. Открывается окно с параметрами графика.

Курсор находится в поле **Заголовок (Title)**;

- Напечатать название графика (например, «количество кур»);
- В первом поле **Variable** нажать кнопку **Sel** и из выпадающего списка выбрать одну из предлагаемых переменных;
- Во втором поле **Variable** выбрать другой элемент модели, например, «количество яиц»;
- Поставить галочку в поле **Scale**, которое находится между теми переменными, которые необходимо сравнить, в данном примере «численность кур» и «количество яиц», то есть поставить галочку между первой и второй строками;

- В поле **Dataset** ввести имя прогона, элемент из которого выбран (у нас «Прогон1»).
- В поле **LineW** (Ширина линии) выбрать ее ширину, например, 2 (Рисунок 59).

Name: GRAPH Quick Hide: ☐ Title ☐ X Label ☐ Legend

Title: Популяция кур и количество яиц

X-Axis: Time Sel X Label:

X-min: X-max: X-divisions: 1 ☐ Lbl-Interval Y-div:

Stamp: Comment:

Type: ☒ Norm ☐ Cum ☐ Stack ☐ Dots ☐ Fill Width: Height:

Scale	Variable	Dataset	Label	LineW	Units	Y-min	Y-max
<input checked="" type="checkbox"/>	численность кур Sel	прогон1		2			
<input type="checkbox"/>	количество яиц Sel	прогон1		2			
<input type="checkbox"/>	Sel						
<input type="checkbox"/>	Sel						
<input type="checkbox"/>	Sel						
<input type="checkbox"/>	Sel						

☐ As WIP Graph (maxpoints) Copy to... Test output ☐ Soft Bounds

OK As Table... Cancel

Рисунок 59 – Настройка графика пользователя.

- Нажать **OK**.
- Нажать кнопку **Display** (Рисунок 40).

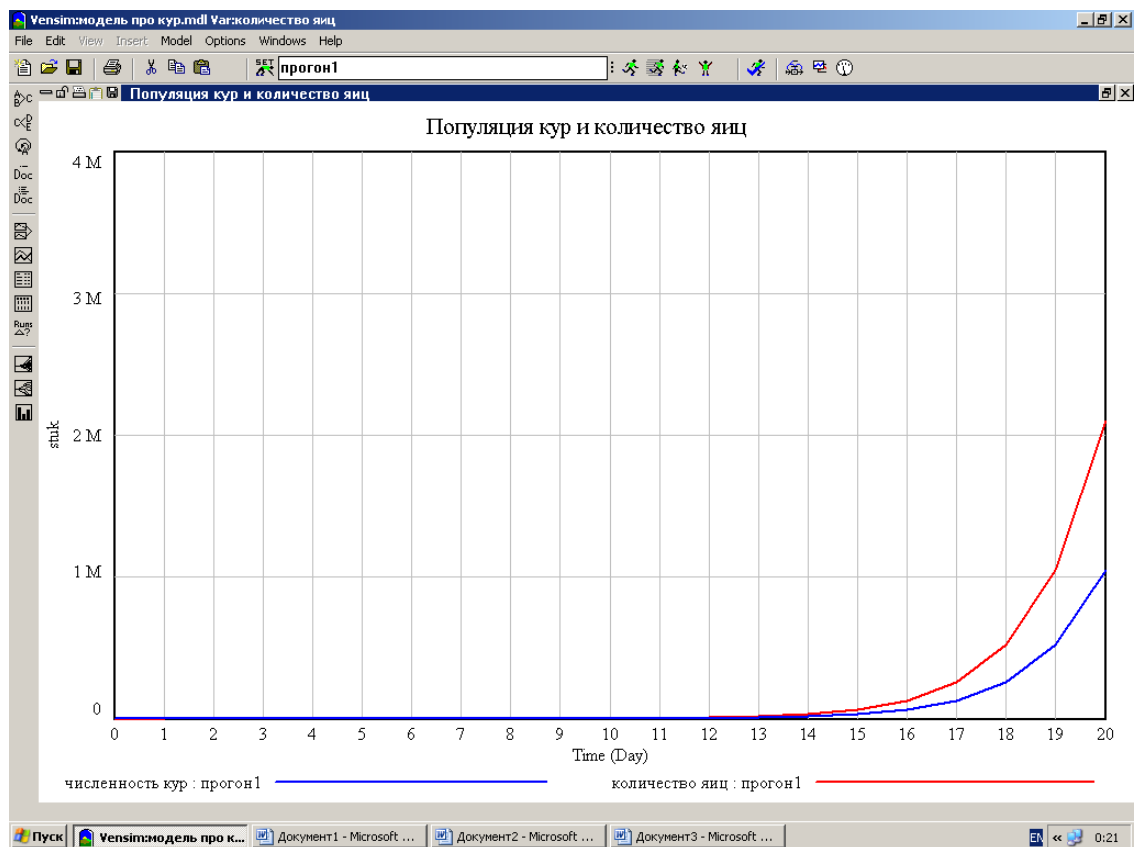


Рисунок 60 – Изображение графика пользователя.

На появившемся графике, например, можно видеть, как поведение одной переменной связано с поведением другой переменной. Если не включать набор данных рядом с именем переменной в диалоге графа, то кнопка **Display** покажет первый загруженный набор данных.

Таблицы пользователя

Для сравнения результатов прогонов можно создать не только графики, но и таблицы пользователя. Для этого нужно выполнить последовательность действий:

- Нажать кнопку **New**. Далее заполнить поля открывшегося окна аналогично тому, как это делалось для построения графика пользователя.
- Далее нажать кнопку **As Table...** и <OK>;
- Нажать кнопку **Display**.

Таблица выглядит следующим образом (Рисунок 61).

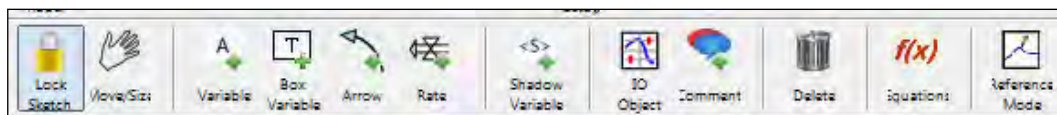
Популяция кур и количество яиц				
Time (Day)	0	1	2	3
численность кур	1	2	4	8
количество яиц	0	2	6	14

Рисунок 61 – Таблица пользователя

По таблице и графику видно, что наблюдается экспоненциальный рост обоих показателей, так как они имеют положительные обратные связи.

3.9. Панель инструментов для построения модели

Программа *Vensim PLE* содержит следующие элементы для создания модели:



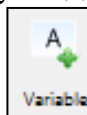
Набор инструментов различен для каждой из версий программы.



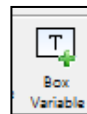
- **(Lock sketch)** используется для выбора элемента модели и блокирования перемещения данного элемента.



- используется для выбора элемента модели и его передвижения по эскизу модели.



(Variable) – ввод вспомогательной переменной в модель.



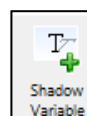
(Box Variable) – ввод переменной, являющейся уровнем.



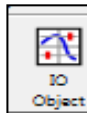
– вставка соединительной стрелки.



- создание темпов.



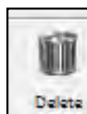
– вставка теневой переменной.



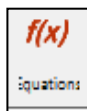
– открывает диалог настроек входных и выходных параметров.



– используется для добавления комментариев.



– удаление элементов модели.



– ввод выражений (формул зависимостей).

Нажатие на кнопку делает данный инструмент до тех пор, пока не будет нажата другая кнопка.

После выбора элемента модели с ним можно производить какие-либо действия путем выбора определенных позиций окна, открывающегося после щелчка по выбранному элементу модели правой кнопкой мыши (Рисунок 62).

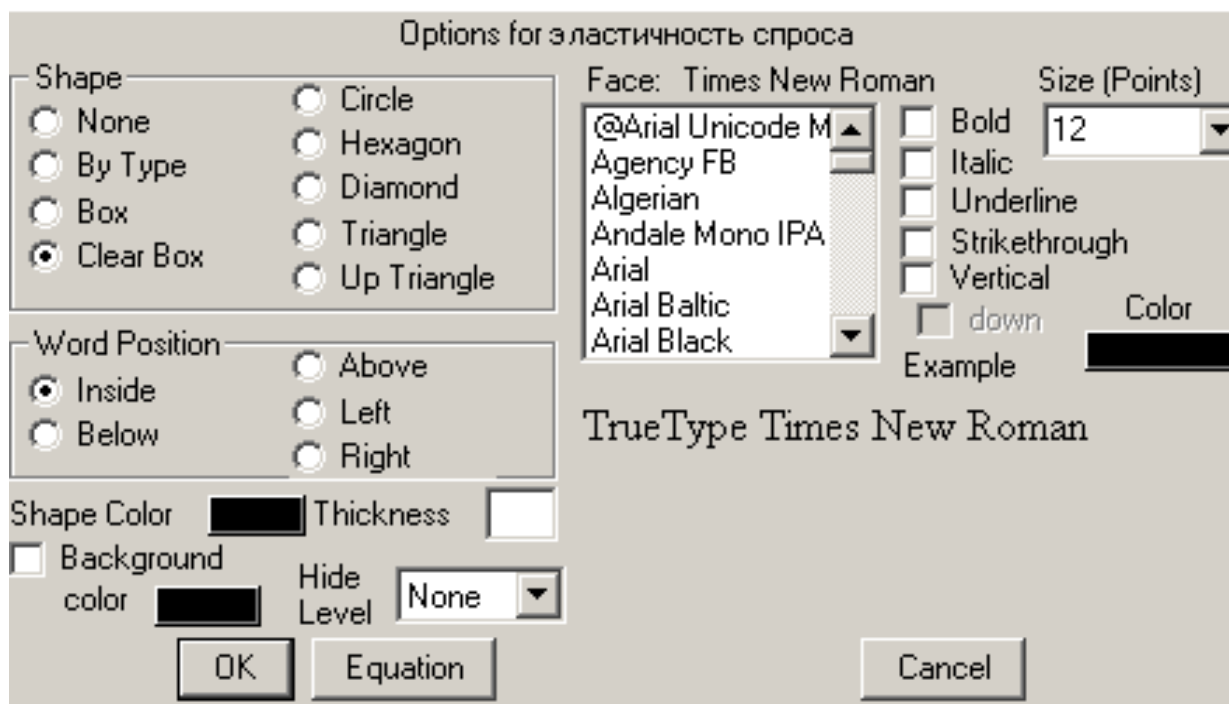


Рисунок 62 – Установление шрифта и цвета выбранной переменной

3.10. Импорт данных в модель

(from .dat format)

Vensim позволяет импортировать данные из файлов формата *.dat. Для этого необходимо предварительно в текстовом файле с расширением .dat, записать данные следующей структурой:

Имя переменной / комментарий

Точка времени 1 Значение 1

Точка времени 2 Значение 2 *комментарий*

Точка времени 3 Значение 3

.....

Точка времени n Значение n

Комментарии являются необязательными, в имени переменной они записываются после знака / в соответствующей строке после пробела.

Точка времени – время, в котором переменная примет **Значение 1**.

Пример приведен на Рисунке 63:

«Убеждаемость / убеждаемость населения в покупке товара	
1	0.01
2	0.03 во втором месяце убеждаемость будет 0,03 чел./контакт
3	0.07
4	0.09
Общаемость / частота общения клиентов и потенциальных клиентов фирмы	
1	25
2	30
3	33
4	35 в четвертом месяце общаемость будет равна 35 контактов в месяц»

Рисунок 63 – Информация программы об импортированных данных

Для того, чтобы загрузить данные в *Vensim* необходимо на панели функций нажать **Model>from .dat format** и выбрать файл с этими данными (Рисунок 64).



Рисунок 64 – Выбор файла с данными для импорта

После чего появится окно, где нужно выбрать *Cyrillic (Windows)*. Это необходимо, чтобы переменные с русскими именами прочитались программой.

Далее после нажатия <Ок> проверяются результаты загрузки данных и выдается информация о количестве загруженных значений (Рисунок 65).

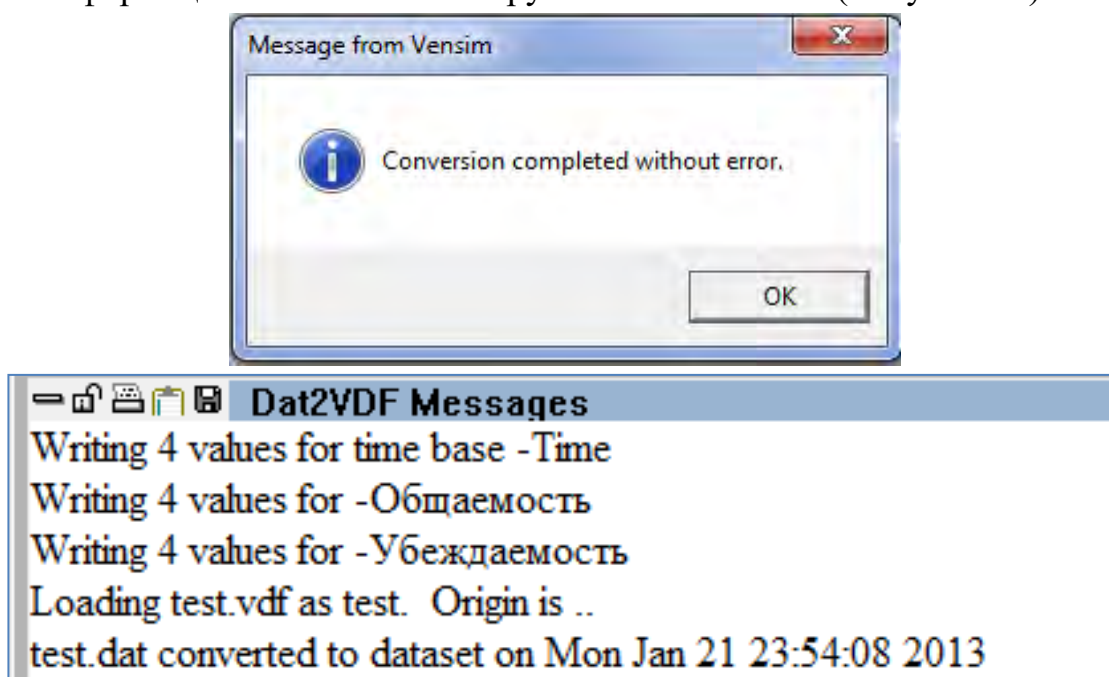


Рисунок 65 – Результаты загрузки данных

По данным в окне видно, что для переменных *Time*, *Общаяемость* и *Убеждаемость* загрузилось по 4 значения.

Проверить загрузку данных можно просмотрев табличные значения *Общаяемости* (Рисунок 66).

Table Time Down		
Time (Month)	"Общаяемость	Общаяемость
1	ь" Runs:	25
2	test	30
3		33
4		35

Рисунок 66 – Проверка загрузки данных.

После импорта данных из .dat-файла, рядом с ним создается прогон с таким же именем.

3.11. Пример построения модели в системе Vensim PLE

Имитационная модель динамики численности населения региона

В данном случае моделируется изменение численности населения региона. Если фиксировать численность населения на начало каждого года, то она

будет определяться как имеющееся количество на начало года плюс разница между родившимися за этот период и умершими:

Численность населения на начало n -го года = численность населения на начало $n - 1$ года + (родившиеся – умершие).

Так как рождаемость неодинакова из года в год, то можно взять усредненный показатель, например, 0,125 чел. в расчете на одного человека населения страны (125 чел./1000 чел.). Следовательно, количество родившихся в данном году будет исчисляться как произведение численности населения на начало года и коэффициента рождаемости:

*Количество родившихся (чел./год) = коэффициент
рождаемости \times численность населения*

Количество умерших за год определяется по следующей формуле:

Количество умерших (чел./год) = Численность населения / Средняя продолжительность жизни

Пояснения: количество родившихся и количество умерших в год в системной динамике можно квалифицировать как **темпы**. Переменная «население» называется основной или **уровнем**. Переменные «коэффициент рождаемости» и «средняя продолжительность жизни» называются дополнительными. В данном случае они будут введены в модель как константы. Среднюю продолжительность жизни примем равной 70 лет, коэффициент рождаемости 0,125 чел. в расчете на 1 чел.

Для создания модели нужно выполнить следующую последовательность действий.

1. Установка горизонта моделирования.

В основном окне выберите **Файл/ Новая модель**. В открывшемся окне Model setting (Рисунок 67) на вкладке **Time Bounds** в поле **FINAL TIME** следует ввести число лет, на которое производится моделирование (например, 3 года), ввести единицу модельного времени, например, месяц, и шаг моделирования, например, один месяц **<ОК>**.

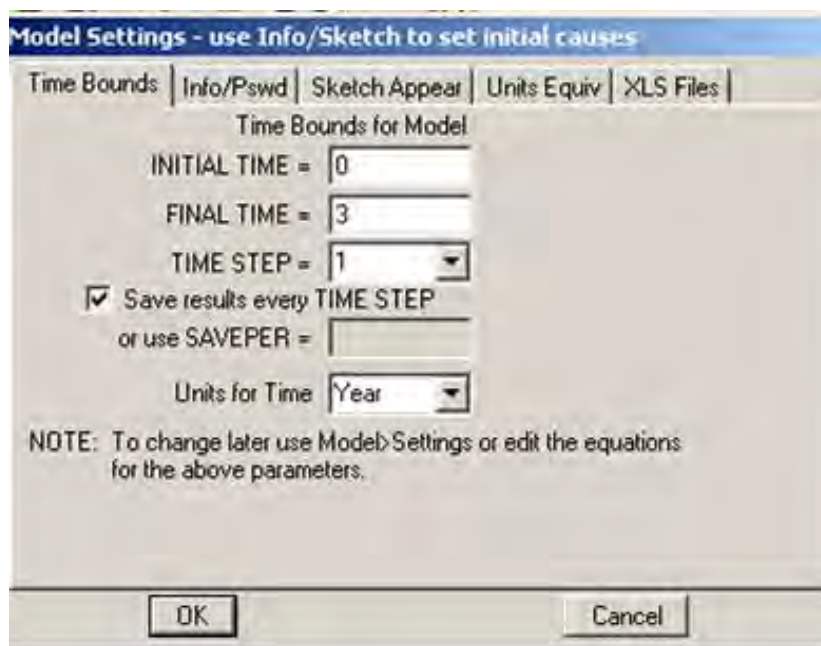



Рисунок 67 – Установка горизонта моделирования

2. Отображение переменных в окне модели.

Выберите **Box Variable**, щелкнув мышью по . Затем расположите курсор примерно посередине окна новой модели и щелкните мышью на этом месте. В появившемся квадрате введите имя переменной и нажмите <Enter> на клавиатуре.


Выберите на панели инструментов **Rate**. Далее, примерно на 5 см левее переменной *Population* в окне моделирования щелкните один раз. После этого щелкните по переменной *Population*. Введите имя темпа *births* (количество рожденных) и <Enter>.

Выберите еще раз **Rate**. Щелкните один раз на квадратике *Population*. Потом щелкните еще один раз правее на 5 см. В открывшемся окне введите имя темпа *deaths* (количество умерших) и нажмите <Enter>.

Выберите **Variable** и щелкните в окне несколько ниже темпа *births*. Назовите переменную *birth rate* (темп рождения) и <Enter>.

Аналогичное сделайте под темпом *deaths*, (в правой части модели) и назовите переменную *average lifetime* (средняя продолжительность жизни).

3. Установление взаимосвязей между уровнями и темпами.

Выберите **Arrow** (стрелки). Щелкните на *birth rate*, проведите курсор в *births* и щелкните снова. (Для того чтобы стрелка приняла удобный для разра-
ботчика модели вид, нужно привести курсор на стрелку до появления знака  и потянуть за этот знак в нужном направлении). Далее щелкните на *average lifetime*, подтяните стрелку к *deaths* и щелкните по этому темпу.

Если необходимо изменить фон и цвет элементов модели, то это делается за заполнением соответствующих строк в **View/Font and Colors**.

В целом имитационная модель роста населения будет выглядеть следующим образом (Рисунок 68).

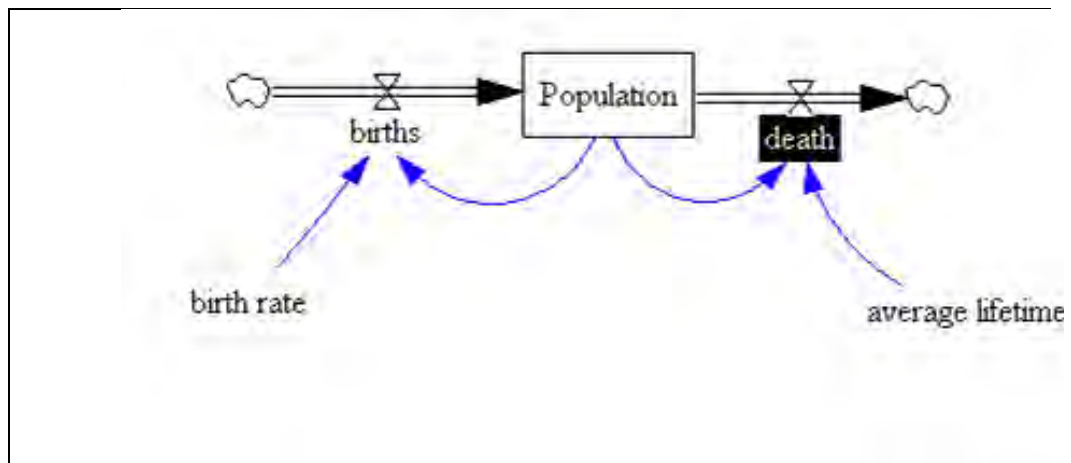



Рисунок 68 – Схема модели

Для каждого элемента модели нужно установить характер его взаимосвязи с другими элементами, темпы и начальные значения уровней. Для этого используется редактор выражений **Equations** (кнопка ).

Далее выберите переменную *birth*, тип переменной *Auxiliary* (дополнительные). Vensim автоматически присваивает темпам и дополнительным переменным тип **Auxiliary**.

Примем следующие обозначения:

Population – численность населения, чел.;

birth rate – коэффициент рождаемости, (чел./год)/чел.;

births – количество рождающихся в год, чел./год;

deaths – количество умирающих в год, чел./год;

average lifetime – средняя продолжительность жизни, лет;

average lifetime – 70 лет.

Тогда описанные выше соотношения будут выглядеть следующим образом:

$birth\ rate = 0.125;$

$average\ lifetime = 70\ \text{лет};$

$births = Population \times birth\ rate;$

$deaths = Population / average\ lifetime;$

$Population = births - death, \text{ начальный уровень} = 140000\ \text{чел.}$

4. Использование редактора выражений **Equations**.

В поле идентификации формулы переменной (в поле знака «=») введите формулу для расчета переменной $birth = Population \times birth\ rate$ (то есть рожденное население определяется как произведение количества населения на темп рождаемости). Это можно сделать, выбрав в этом же окне вкладку **Variables**. Здесь, используя панно калькулятора и щелкнув мышью на высветившиеся в окне **Choose Initial Variable** соответствующие названия переменных, определить эту зависимость.

Далее нужно добавить единицы измерения для переменной *births*. Это делается в поле **Units**. Там нужно установить *selovek/year* (человек в год).

Если в формулах есть синтаксические ошибки, и структура модели не согласуется с редактором выражений, то будет выведено соответствующее сообщение об ошибках.

Кроме типа переменных **Auxiliary** может быть использован тип переменных **Level**. Например, если в окне модели выбрать переменную *Population*, то видно, что Редактор выражений присвоил ей тип **Level** (основная переменная или уровень). В поле **INTEG** нужно отразить, как определяется эта переменная. Численность населения прирастает на величину, равную количеству рожденных за определенный период (год) за минусом умерших за этот же период (то есть $Population = births - deaths$).

Для переменных-уровней **Equation Editor** имеет дополнительное поле **Initial Value** (начальный уровень), где устанавливается начальное значение переменной на нулевой момент времени. Поэтому присвоим этой переменной значение уровня, равное, например, 140000. Единица измерения – количество человек (Рисунок 69).

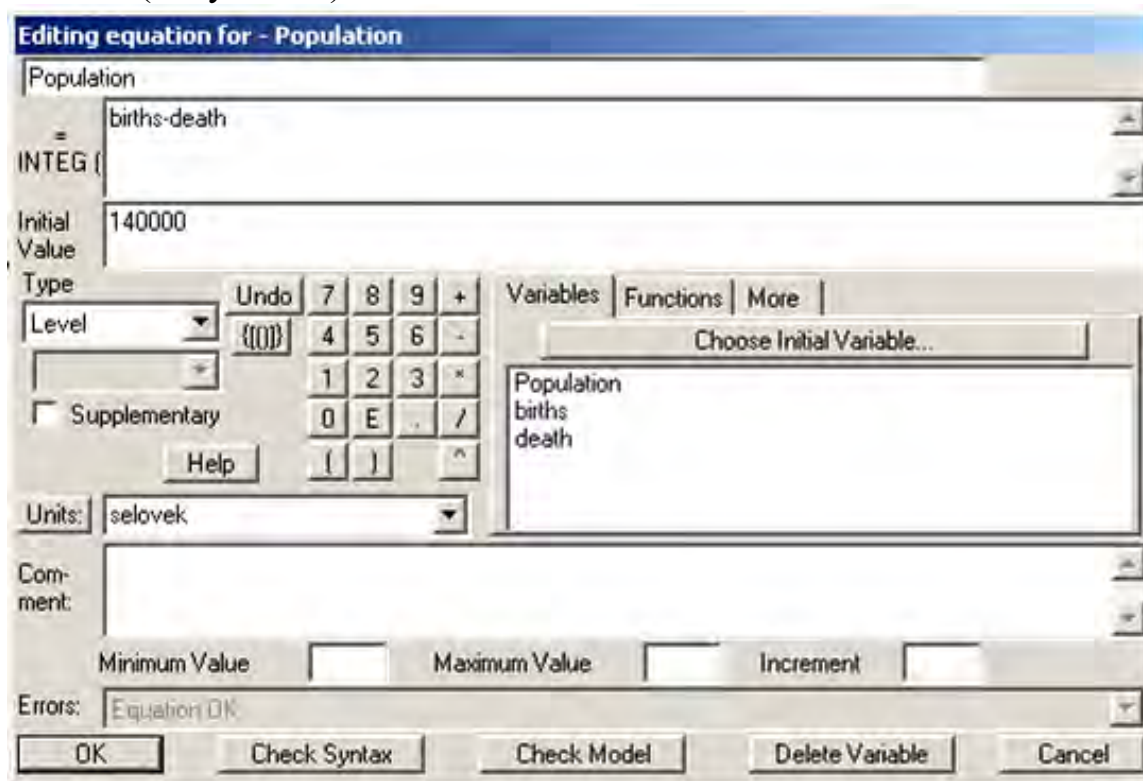


Рисунок 69 – Окно Редактора выражений для переменной *Population*

Нажмите на *birth rate*. Ее тип – константа, единицы измерения - человек в год, количественное выражение примем равное 0,125 (Рисунок 70).

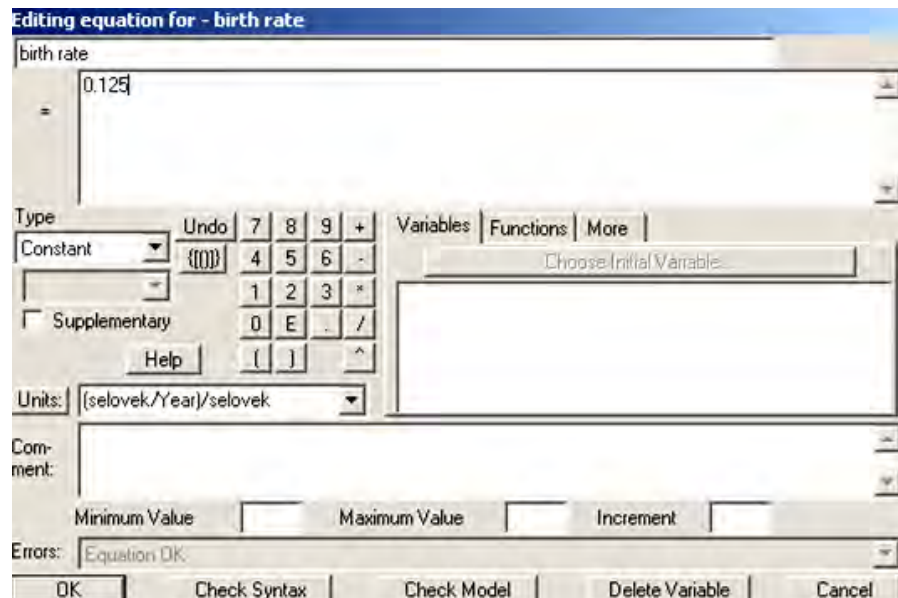


Рисунок 70 – Окно Редактора выражений для темпа *birth rate*

Для средней продолжительности жизни (*average lifetime*) окно выражений будет выглядеть так (Рисунок 71).

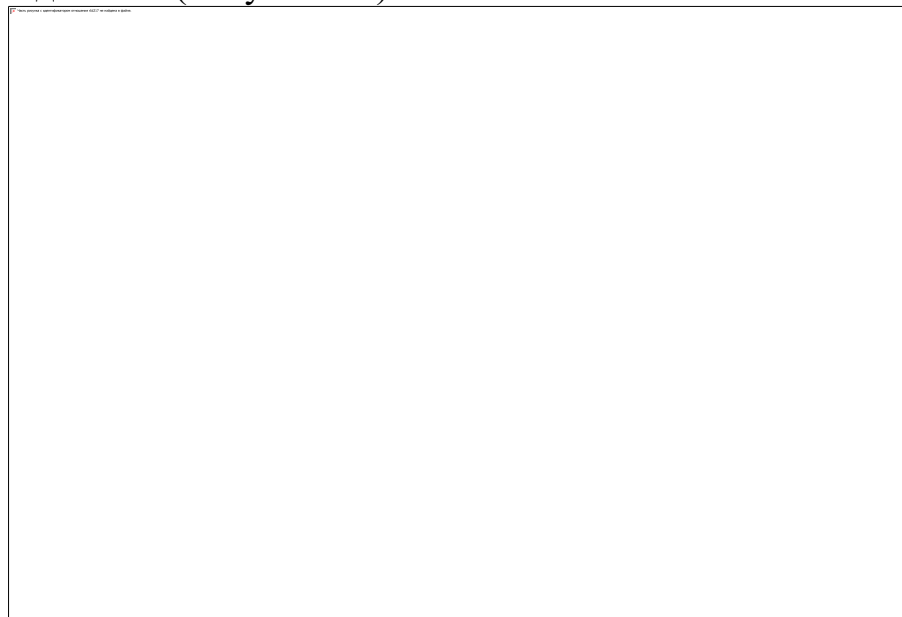


Рисунок 71 – Окно Редактора выражений для дополнительной переменной *average lifetime*

Окно Редактора выражений для переменной *deaths* выглядит следующим образом (Рисунок 72).

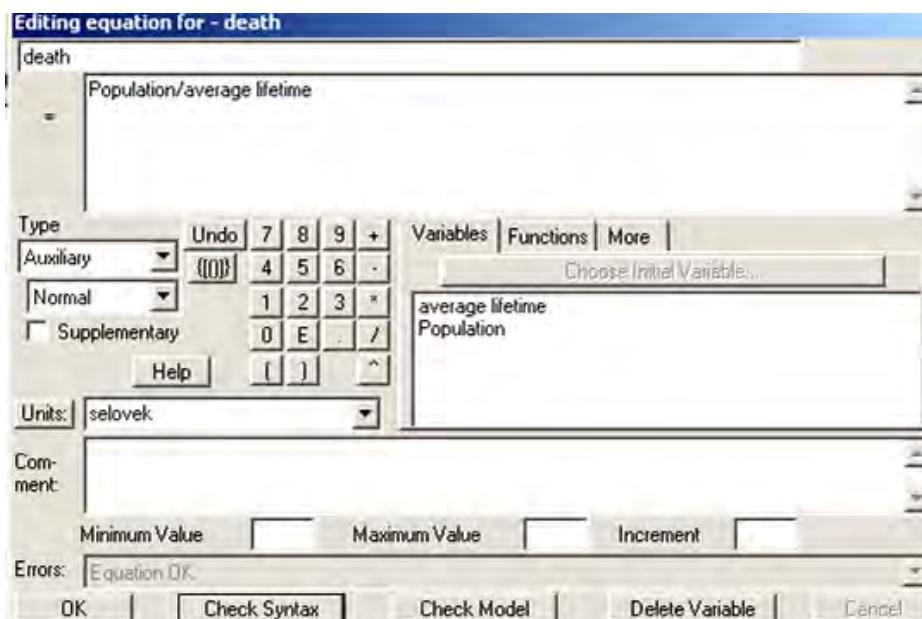


Рисунок 72 – Окно Редактора выражений для *death*.

5. Проверка модели на наличие синтаксических ошибок.

Перед прогоном имитационной модели ее нужно проверить на наличие ошибок в выражениях и установленных единицах измерения. Это делается с помощью выбора **Model/Check Model**. Если ошибок нет, то выдается сообщение «Model is OK». Если в модели имеются ошибки, то редактор выражений укажет на них.

При выборе **Select Model/Units Check** Редактор выражений проверяет, везде ли правильно установлены единицы измерения. При наличии ошибок может быть выдано, например, следующее сообщение (Рисунок 73).

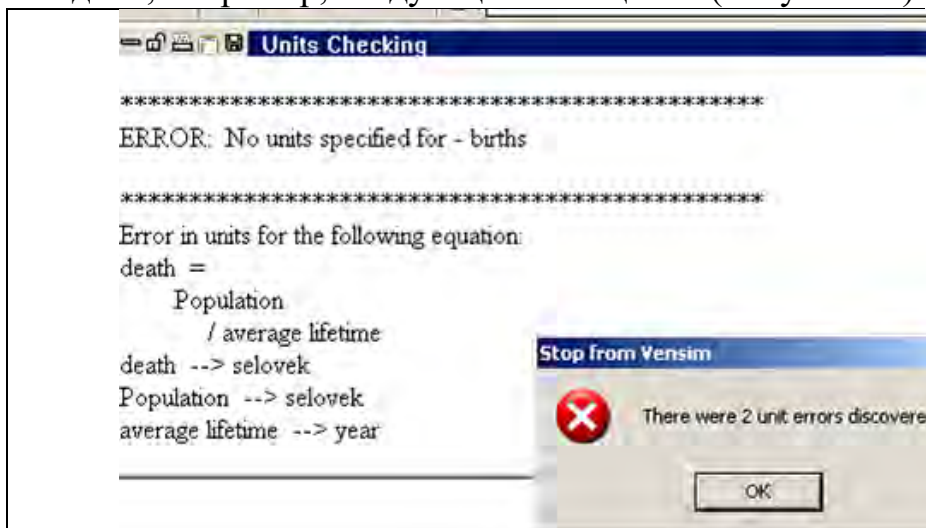

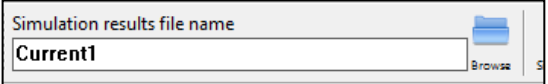



Рисунок 73 – Окно результатов проверки модели


На Рисунке 73 в окне сообщения об ошибке указано, что не установлено единиц измерения для переменной *birth*, и неправильно установлены единицы измерения для переменной *deaths* (вместо «selovek» нужно «selovek/year»).

6. Прогон имитационной модели.

Щелкните на кнопку  бокса . В открывшемся окне укажите имя файла, в котором будут храниться результаты

прогона модели, например, Naselen. Затем щелкните по кнопке  на панели инструментов. При этом Vensim PLE проведет процесс моделирования (прогон модели) и сохранит выходные значения для набора данных Naselen. Если затем требуется изменить какие-либо характеристики модели, то результаты нового прогона можно будет в последствии сравнить с выходными данными Naselen. Этот набор выходных данных сохранил поведение каждой переменной модели на протяжении установленного исследователем горизонта (периода) моделирования.

Анализ результатов моделирования

Имитационная модель создается с целью показать изменение темпов роста населения. Vensim позволяет провести анализ изменения каждой из существующих в модели переменных. Например, при необходимости проследить в динамике поведение основной переменной *Population*, можно щелкнув по ней и затем воспользовавшись инструментом анализа, расположенным на вертикальном панно . Появится окно графического анализа этой переменной (Рисунок 74).

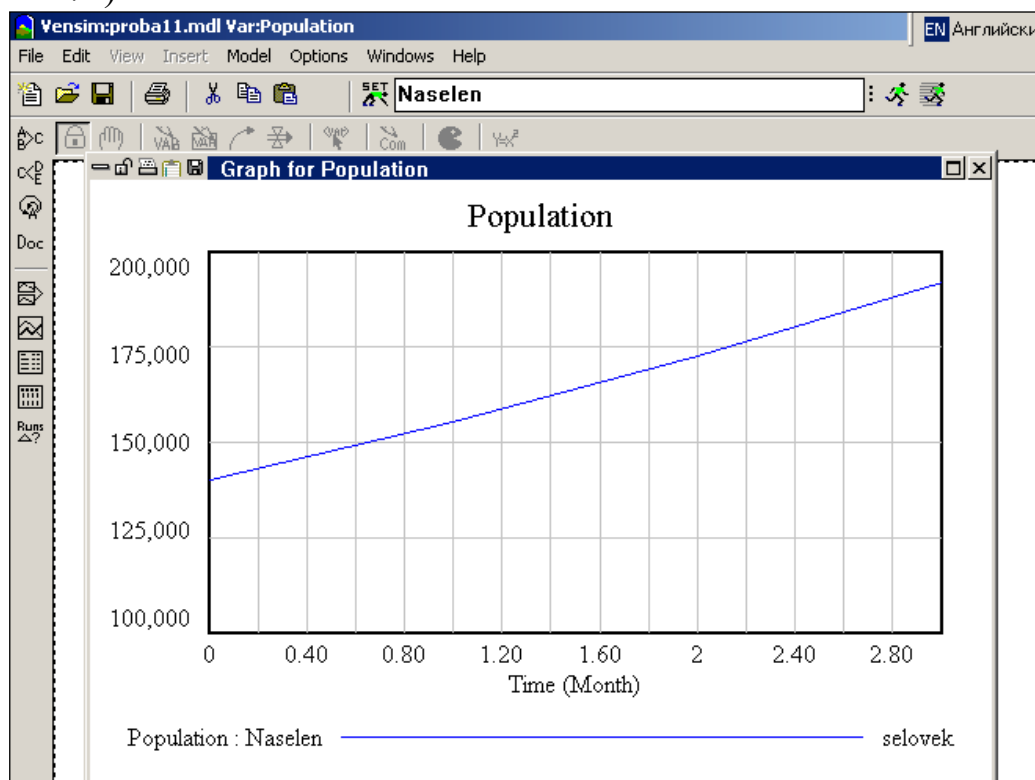


Рисунок 74 – График уровня *Population*

То же самое можно сделать для любой переменной.



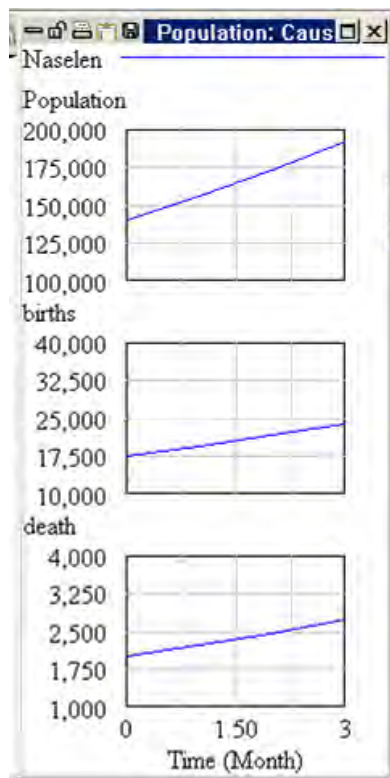


Для анализа можно использовать табличную форму представления данных, щелкнув на инструмент анализа «Table»  (Рисунок 75).

Table			
Time (Month)	0	1	2
"Population" Runs:	Naselen		
Population	140000	155500	17

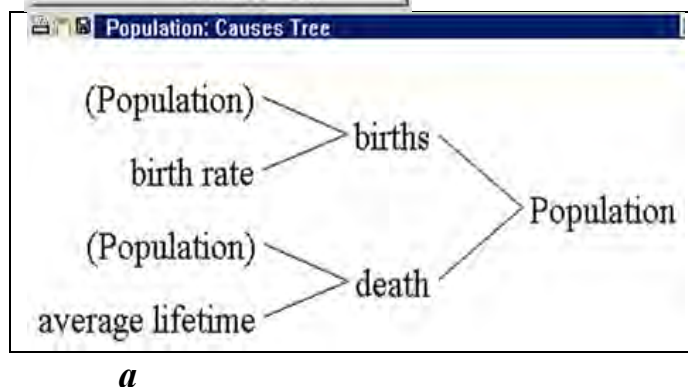
Рисунок 75 – Результаты прогона в табличной форме

Если щелкнуть на инструмент анализа «Causes Strip» , то будет показан полосный граф, отражающий как изменение в динамике значений самой переменной, так и других переменных, непосредственно на нее влияющих.

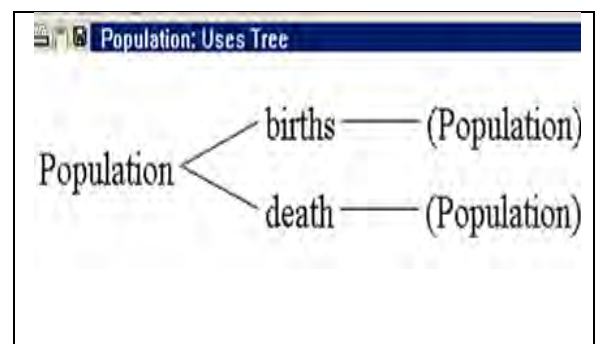


Инструменты анализа позволяют исследовать структуру модели с помощью инструмента анализа «Causes Tree» (Дерево причин)  и «Uses Tree» (Дерево следствий)  (Рисунок 76).

Дерево причин показывает, какие переменные определяют переменную *Population*, а Дерево следствий показывает (до 2-х связей), где используется эта переменная в модели (что она является причиной изменения двух переменных). Эта форма делает части модели более легкими для восприятия.



a

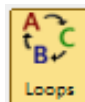


б

Рисунок 76 – Дерево причин (*a*) и дерево следствий (*б*)

Циклы в модели

Только переменные типов **Auxiliary** и **Level** имеют петли обратной связи (циклы), через них проходящие. Так, например, для более четкого представления, какие обратные связи имеет основная переменная *Population*, мож-



но воспользоваться инструментом «Loops». Окно результатов выглядит следующим образом (Рисунок 77).

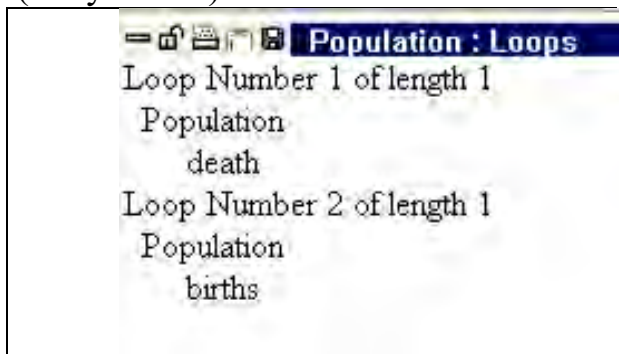


Рисунок 77 – Окно результатов при использовании инструмента «Loops»

Окно показывает, что у переменной *Population* существуют обратные связи с двумя переменными *births* и *deaths*, находящимися от этой переменной на расстоянии одной связи (длины).

Окно результатов, показывающее уравнение, лежащее в основе всех переменных в модели, единицы измерения для переменных и введенные комментарии (если они были введены ранее) открывается при использовании инструмента анализа «Document» (Рисунок 78).

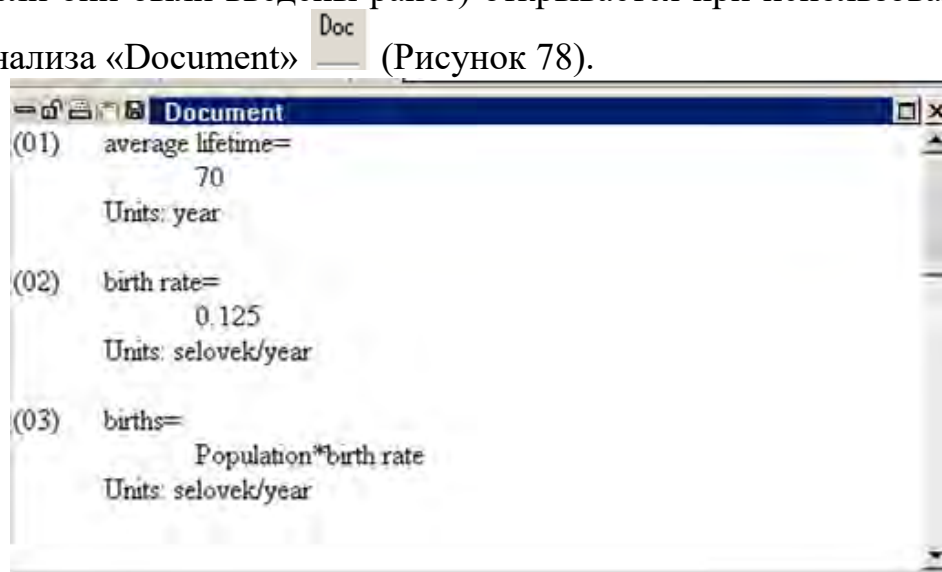



Рисунок 78 – Отображение выражений модели

7. Вычислительный эксперимент.

Система Vensim позволяет сравнивать несколько вариантов выходных характеристик прогонов имитационных моделей, чтобы проследить динамику моделируемых событий при различных вариантах (или стратегиях) развития. Исследователь может вводить различные значения переменных типа Constanta. Чтобы быстро найти эти переменные в модели в Vensim на основной панели инструментов существует специальный инструмент - **Set Up Simulation** . При его активизации переменные типа Constanta появляются жел-

тым текстом на синем фоне. Их значения можно менять, щелкнув по ним и изменив значение в EditBox и <Enter>.

В рассматриваемой модели есть две такие переменные - *birth rate* и *average lifetime*. Изменим их значения. Для *birth rate* поставим 0,08 вместо 0,125, а для *average lifetime* поставим 65 лет вместо 70 лет. Осуществим прогон модели, результаты которой запишется в новый файл Naselen1. Поскольку результаты предыдущих прогонов также сохраняются, то Граф-инструмент при его активизации строит два графика (Рисунок 79). Они показывают, что при сокращении рождаемости и уменьшении средней продолжительности жизни темп роста населения снижается.

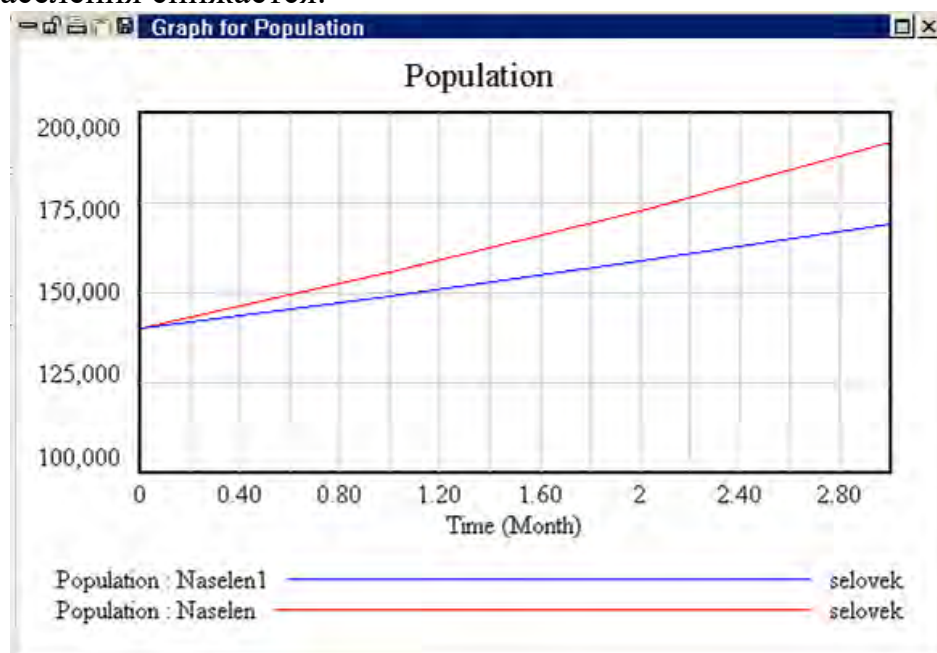


Рисунок 79 – Графическое отображение результатов моделирования

3.12. Функция LOOKUP

Lookup позволяет ввести функцию пользователя без ее математической записи, а только с помощью указания точек на плоскости:

$$y = my\ function(x)$$
$$y = \text{моя функция} (x)$$

Как правило, этот инструмент используется при отражении нелинейных функций. Например, если изменить условия модели, приведенной выше и заменить переменную «население» на «численность зайцев», логика прироста их численности несколько отличается от приведенной в модели выше.

В частности будет неправильным не учитывать факторы, которые в условиях окружающей среды препятствуют неограниченному их размножению. Это, например, наличие корма, территория и другие (назовем их условиями окружающей среды).

То есть рост зайцев представляет собой нелинейную зависимость от времени.

По отношению к модели, приведенной выше с численностью населения переменная *Умершие* изменяется по отношению к переменной *Население* линейно (Рисунок 80).

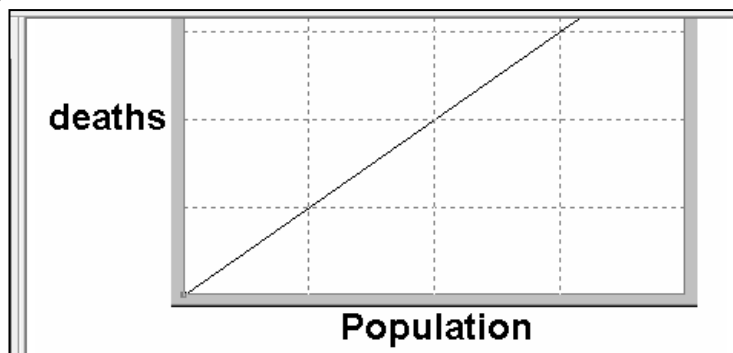


Рисунок 80 – Относительное изменение переменной *Умершие* (*death*)

Это не значит, что переменная *death* будет линейно возрастать с течением времени. Линейность означает, что переменная *death* будет расти так же, как переменная *Population* (если *Population* растет в геометрической прогрессии, так же будет расти и *death*).

Для выражения зависимостей переменных модели не от времени, а по отношению к другим переменным, используется функция *Lookup*. Можно было бы сделать таблицу *lookup*, которая выражает эту линейную зависимость, но при линейной зависимости проще всего это написать формулой в редакторе выражений. Формула предполагает выражение *Population / average lifetime* так, что наклон прямой будет $1 / \text{average lifetime}$.

А в задаче с численностью зайцев необходимо установить, чтобы переменная *Умершие в год* увеличивалась быстрее, чем переменная *Численность зайцев*, так как если бы *Численность зайцев* увеличивалась линейно, то она быстро бы достигла неадекватно до больших размеров (так как количество кормов и других ресурсов ограничено, и, следовательно, зайцы будут умирать быстрее).

В данном случае нужно построить, например, такое соотношение (Рисунок 81).

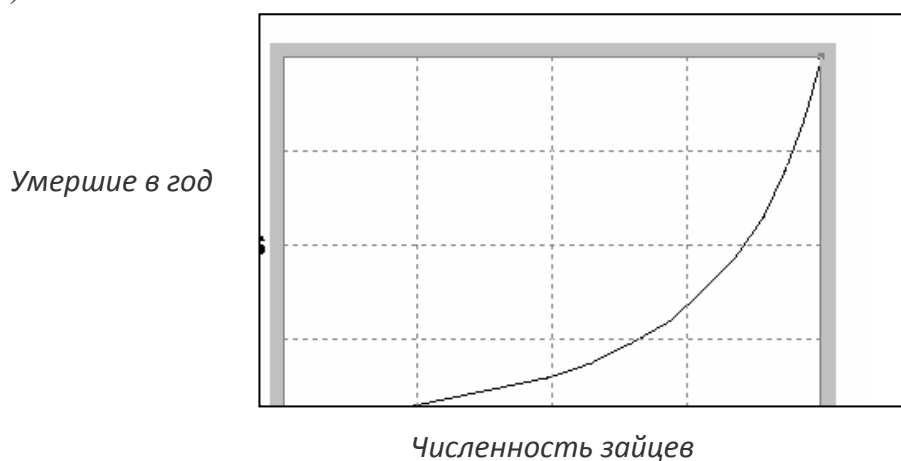


Рисунок 81 – Нелинейная зависимость между переменными

Это можно сделать с помощью Lookip. Но вначале нужно нормализовать данные. Они могут иметь разную размерность или одну и ту же, но сильно различающуюся по значению.

Далее условимся называть тот элемент модели, от которого идет стрелка (то есть который задействован в формуле), входом, а тот элемент, к которому идет стрелка (то есть, зависящего от первого элемента) выходом.

Нужно построить нормированный вход вокруг ориентировочных точек с координатами 0,0 и 1,1. Вход устанавливается безразмерным и не зависит от единиц измерения и масштаба других переменных в модели. Выход также безразмерный и также не зависит от единиц измерения и масштабов других переменных (так как предварительно проведена нормализация значений). Например, предположим, что мы должны были измерить *Численность зайцев* в тысячах зайцев (напомним – это уровень), а *Умершие в год* в тысячах зайцев в месяц (так как это – темп). Функция *Lookip*, берущая *Умершие в год* в качестве входных данных и имеющая *Умершие в год*, как выход, уже будет недействительной без нормализации, так как у них разные единица измерения.

Нормализация позволяет достичь желаемых поведенческих взаимосвязей с обобщенным набором чисел в функции Lookip. Если сведения о размере и характеристиках численности зайцев изменились, можно просто изменить значение переменной *Возможность окружающей среды* или переменной *Средняя продолжительность жизни*, а не всю функцию *Lookip*. Схема модели с введенными элементами приведена на Рисунок 82.

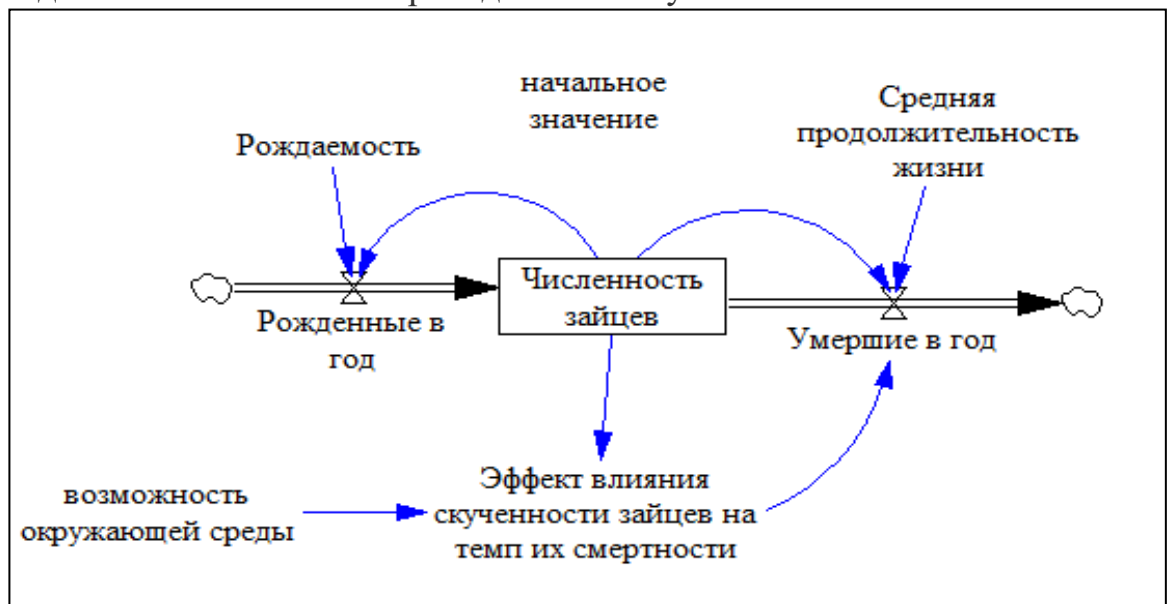


Рисунок 82 – Схема модели с новыми введенными элементами

Для нормализации нужно разделить входную величину на ее нормальное или среднее значение (например, *Возможность окружающей среды*). Ко-

гда фактическое значение переменной *Численность зайцев* равно его нормальному значению, вход (в функции *Lookup*) равен 1. Другие значения *Численность зайцев* будут изменять вход выше или ниже единицы. Выходные значения *Lookup* также будут варьировать вокруг единицы. Выход *Lookup* может потом быть использован для определения значения другой переменной модели выше или ниже ее нормального значения. Другими словами: если входная переменная равна его нормальному или среднему значению, выход в функции *Lookup* равен 1 и, следовательно, не оказывает влияния на связанную выходную переменную.

С другой стороны, функция *Lookup*, нормализованная с использованием *Численности зайцев* по отношению к *Возможности окружающей среды* служит входом для выхода *Эффект влияния скученности зайцев на темп из смертности*, которая, в свою очередь, является входом для переменной *Умершие в год*, которая рассчитывается как (*Численность зайцев/Средняя продолжительность жизни*). И нормализация не должна повлиять на изменение их единиц измерения (Рисунок 83).

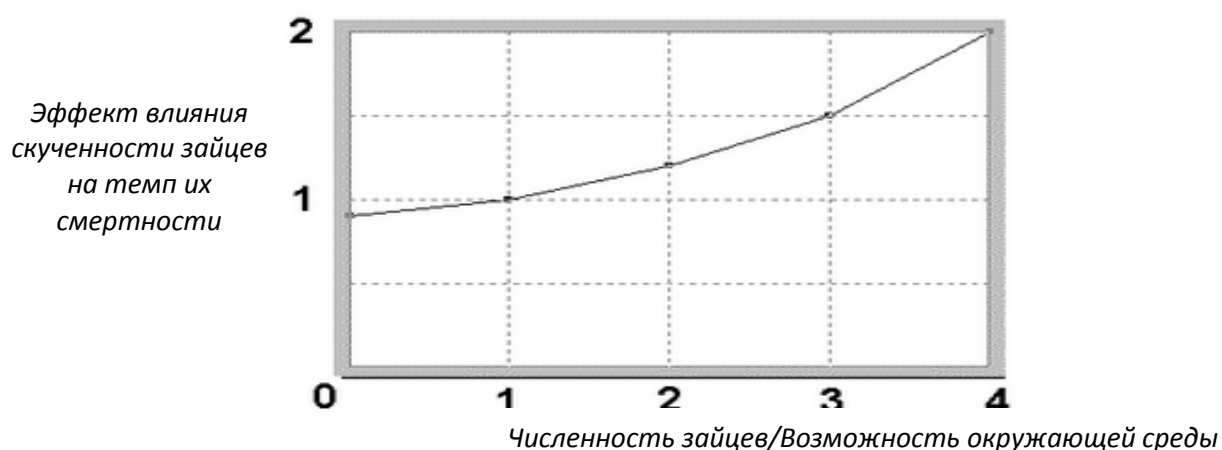


Рисунок 83 – Графическое отображение нормализованных величин

Пример. Для ее построения выберите новую модель и установите FINAL TIME = 30 годам, шаг = 0,125 года. Далее нужно нарисовать эскиз модели (Рисунок 84).

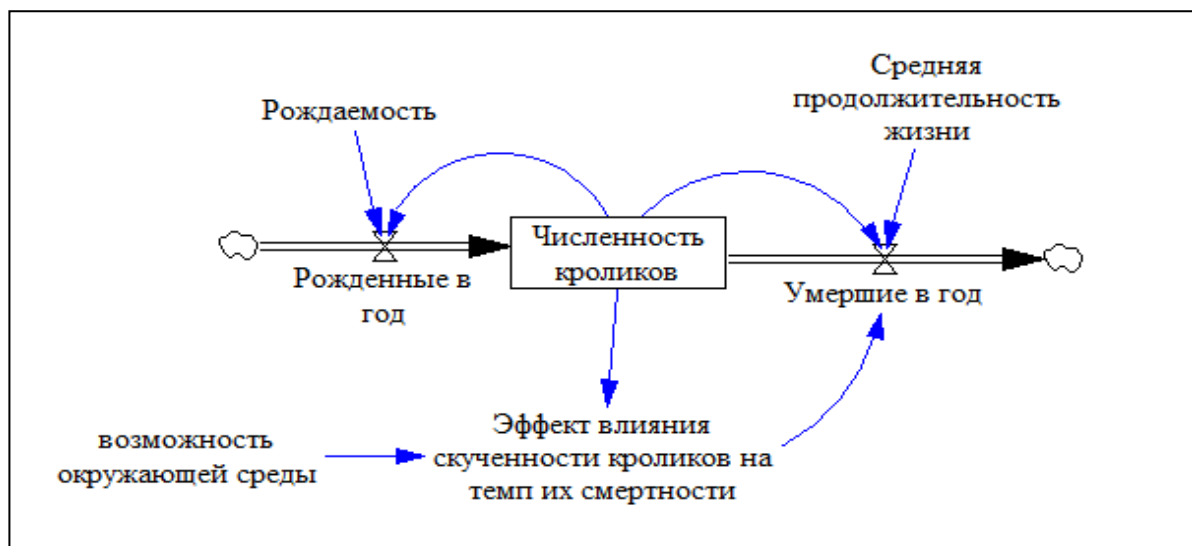


Рисунок 84 – Модель с добавленными переменными

Далее с помощью редактора уравнений определить следующие взаимосвязи (Рисунок 85).

- (01) FINAL TIME = 30
Units: Year
The final time for the simulation.
- (02) INITIAL TIME = 0
Units: Year
The initial time for the simulation.
- (03) SAVEPER = TIME STEP
Units: Year [0,?]
The frequency with which output is stored.
- (04) TIME STEP = 0.125
Units: Year [0,?]
The time step for the simulation.
- (05) Рождаемость = 0.23
Units: доля/Year
- (06) Рожденные в год = Численность зайцев * Рождаемость
Units: зайцев/Year
- (07) Средняя продолжительность жизни = 8
Units: Year
- (08) Умершие в год = (Численность зайцев / Средняя продолжительность жизни) * Эффект влияния скученности зайцев на темп их смертности
Units: зайцев/Year
- (09) Численность зайцев = INTEG (Рожденные в год - Умершие в год, начальное значение)
Units: зайцев
- (11) возможность окружающей среды = 1000
Units: зайцев
- (12) начальное значение = 1000
Units: зайцев.

Рисунок 85 – Выражения модифицированной модели

Для установления выражения для переменной *Эффект влияния скученности зайцев на темп их смертности* с использованием функции *Lookup* нужно произвести следующие действия:

- в редакторе уравнений выделить данную переменную;
- выбрать тип *Auxiliary*, а подтип, который по умолчанию будет определен как *Normal*, заменить на **with Lookup**, появляющийся из выпадающего списка. Это несколько поменяет поля окна редактора формул – будет введено новое поле *Look up* (оно впоследствии при его заполнении будет заменено на поле *Initial Value*), а справа от поля подтипа переменной появится кнопка *As Graph*;
- в поле *Variables* ввести выражение *Численность зайцев/ возможность окружающей среды*;
- установить курсор в поле *Look up*. Затем нажать на кнопку *As Graph*. В открывшемся окне ввести в левое поле *New* цифру 1 и <Enter>, курсор переместится в правое поле. В правом поле набрать 0.9 и <Enter>, обе цифры встанут в первую позицию полей *Input* и *Output*. То же самое сделать с оставшимися четырьмя парами цифр. Одновременно можно видеть рисуемый график (Рисунок 86);
- установить максимальные значения X и Y соответственно 4 и 2.

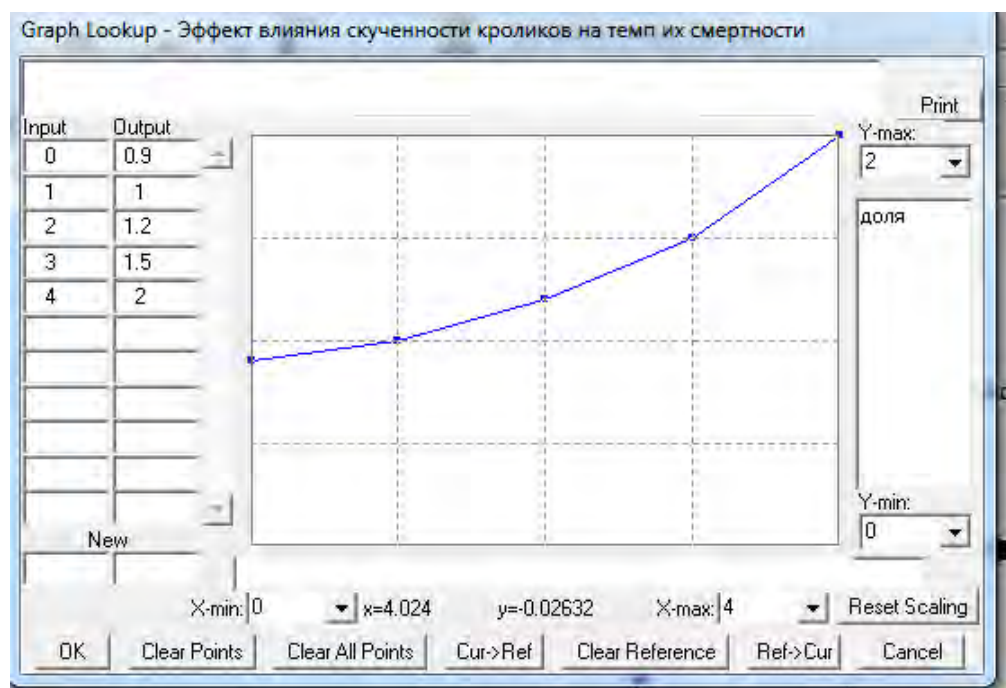


Рисунок 86 – Использование функции *Lookup*

- нажать OK, при этом и автоматически будет заполнено поле *Look up*, которое будет уже называться *Initial Value* (Рисунок 87).

Equations	Численность кроликов/возможность окружающей среды
Initial Value	([(0,0)-(4,2)], (0,0.9), (1,1), (2,1.2), (3,1.5), (4,2))

Рисунок 87 – Заполнение поля *Look up*

Таким образом, переменная будет вычисляться так (Рисунок 88).

Эффект влияния скученности зайцев на темп их смертности =
 WITH LOOKUP (Численность зайцев/возможность окружающей
 среды, $[(0,0)-(4,2)], (0,0.9), (1,1), (2,1.2), (3,1.5), (4,2))$,
 Units: Dmnl

Рисунок 88 – Запись выражения с помощью функции *Lookup*.

Провести прогон модели, сохранив его под каким-либо именем, например, «Зайцы база».

Просмотрим результаты с помощью инструментов аналитической панели (Рисунок 89).

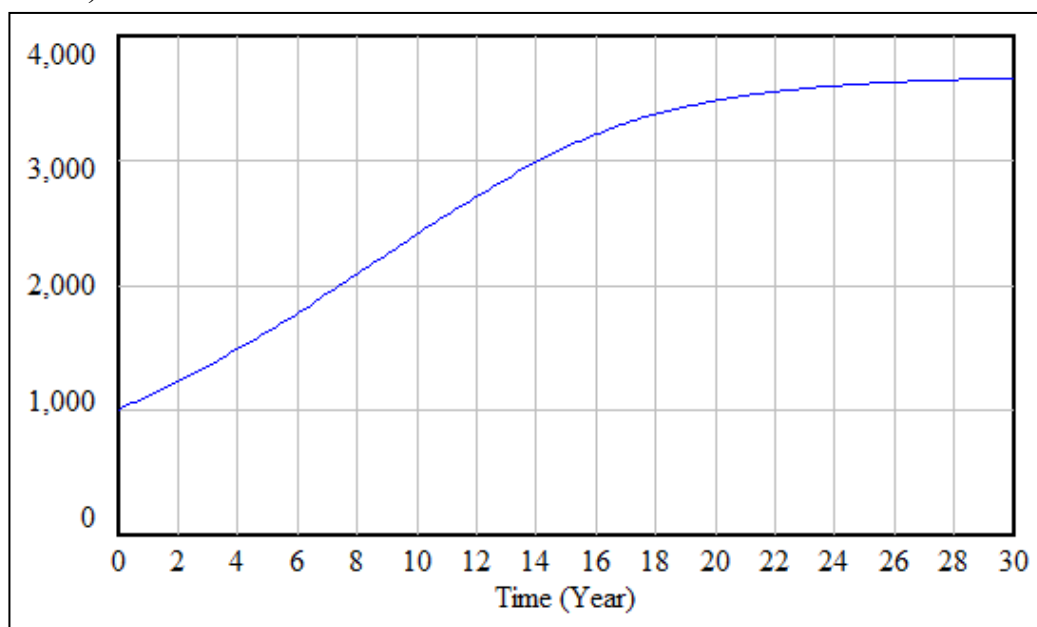


Рисунок 89 – Нелинейность изменения переменной во времени

На графике видно, что максимальная численность зайцев будет около 3500. Это – из-за тех условий, которые были введены нами с помощью функции *Lookup*.

3.14. Оценка чувствительности моделей системной динамики (доступна в версиях от Vensim PLE Plus и выше)

Чувствительность модели к изменению определенных ее элементов определяется степенью реакции на это изменение итоговых характеристик модели.

Чувствительность имитационной модели определяется на основе метода Монте-Карло, так как система в данном виде моделирования представляется в виде «черного ящика», исследование состояния которого производится оценкой реакции системы на импульсы (воздействие факторов). Для оценки чувствительности методом Монте-Карло программой производится случайный выбор значения в пределах выбранного исследователем интервала по одному или сразу нескольким входным параметрам (факторам) определенного значения. Далее производится прогон модели, и фиксируются значения одного или нескольких результативных показателей. Поскольку выбор конкретной величины факторного признака из диапазона значений производится случайным способом, то и полученная после каждого прогона величина результативного признака является случайной величиной. Множество прогонов (несколько десятков или даже сотен) формирует выборочную совокупность значений результативного признака. Значения результативного признака распределены определенным образом. Характер распределения может быть различным. Программа Vensim результаты распределения выдает в виде графика, на котором.

Распределение случайной величины внутри интервала может быть установлено как равномерное или как неравномерное. При равномерном распределении (RANDOM_UNIFORM) принимается, что появление любого значения внутри интервала равновероятно. Среди неравномерных распределений чаще всего используется нормальное распределение (RANDOM_NORMAL). Кроме него, в программу Vensim встроены такие виды неравномерного распределения как распределение Пуассона (RANDOM_POISSON), Вейбулла (RANDOM_WEIBULL), Триангулярное (RANDOM_TRANGULAR), Биномиальное (RANDOM_BINOMAL), Гамма (RANDOM_GAMMA) и другие. Они открываются с помощью кнопки Distribution в мастере оценки чувствительности.

Например, откроем модель HOUSES.MDL (Рисунок 90).

Если нет четких представлений о примерном распределении величины, то используют равномерное распределение. Далее в программе задаются максимальные и минимальные границы по выбранным параметрам.

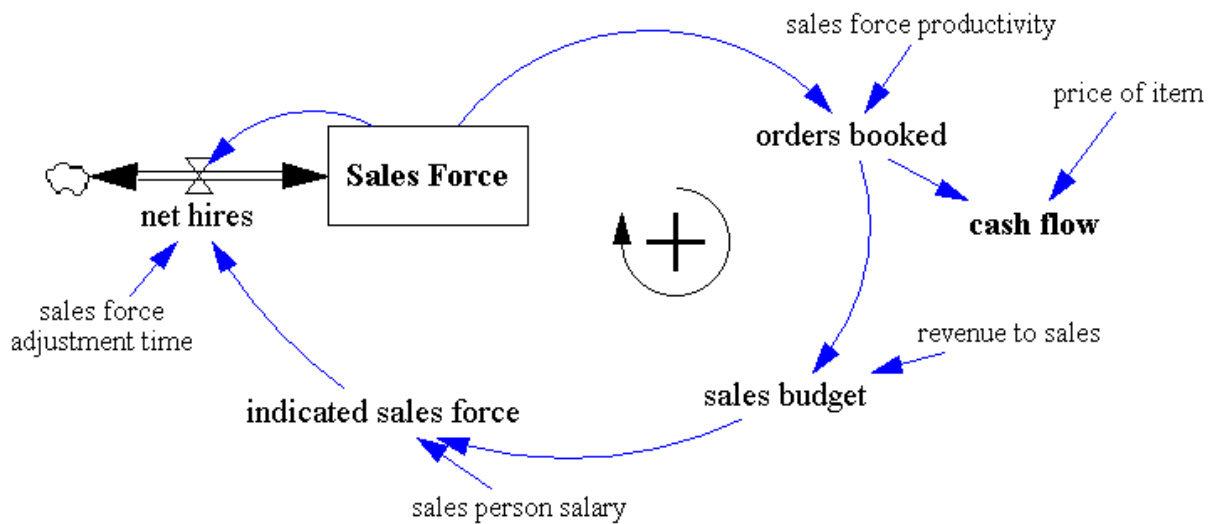



Рисунок 90 – Схема модели HOUSES.MDL

В программе *Vensim* чувствительность каких-либо параметров модели в каком либо входным факторам оценивается с помощью встроенной процедуры. Она запускается одним из двух способов: нажатием на кнопку  на панели инструментов или запуском мастера оценки чувствительности с помощью команды основного меню *Model/Sensitivity* (Рисунок 91).

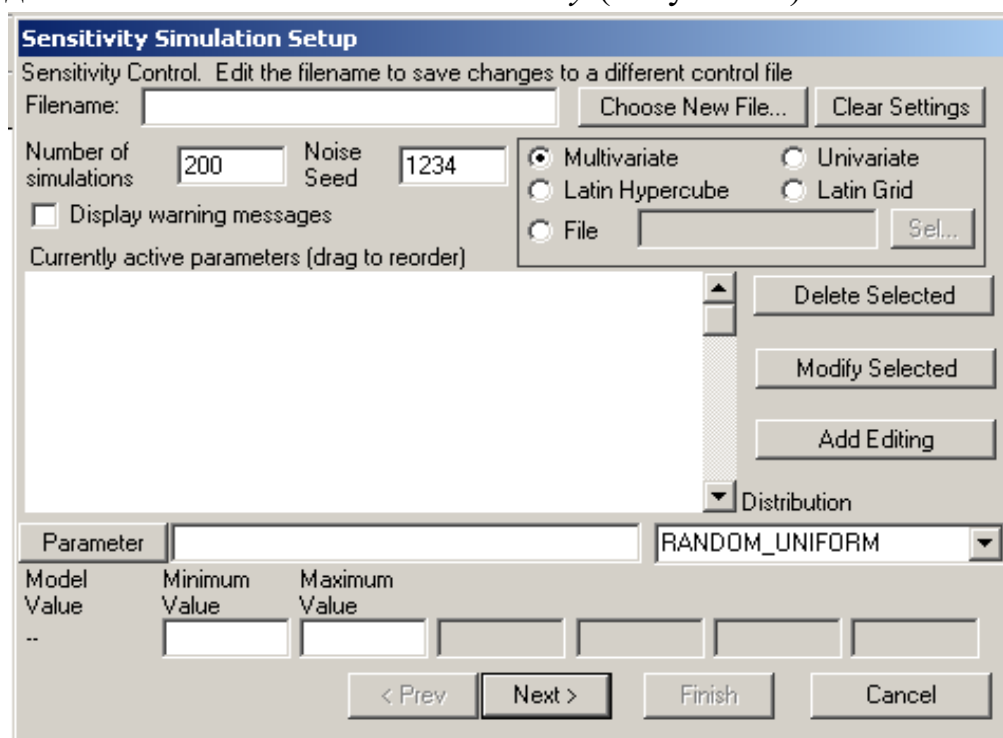


Рисунок 91 – Окно настроек анализа чувствительности

В рассматриваемой модели установим следующие диапазоны изменения сразу для трех констант, учитывая, что в качестве отклика взята переменная (уровень) *Sales Force*.

- установим количество прогонов 200;
- поставим флажок для многомерного выбора (изменение несколько параметров) – *Multivariate*;
- выберем изменяемую константу из выпадающего списка после нажатия клавиши Parameter. Первая изменяемая константа - *sales force productivity*;
- установим минимальное и максимальное значение для данной переменной – соответственно 200 и 220;
- выберем равномерное распределение RANDOM_UNIFORM;
- нажмем кнопку Add Editing – переменная добавится в поле выбранных переменных.

Аналогичные действия произведем с переменной *sales person salary*. Установим для нее нормальное распределение на интервале от 1800 до 2200. При нормальном распределении активными становятся два поля, касающиеся характеристик нормального распределения. Заполним поля *Mean* и *Standard Deviation* соответственно цифрами 2000 и 100 (Рисунок 92).

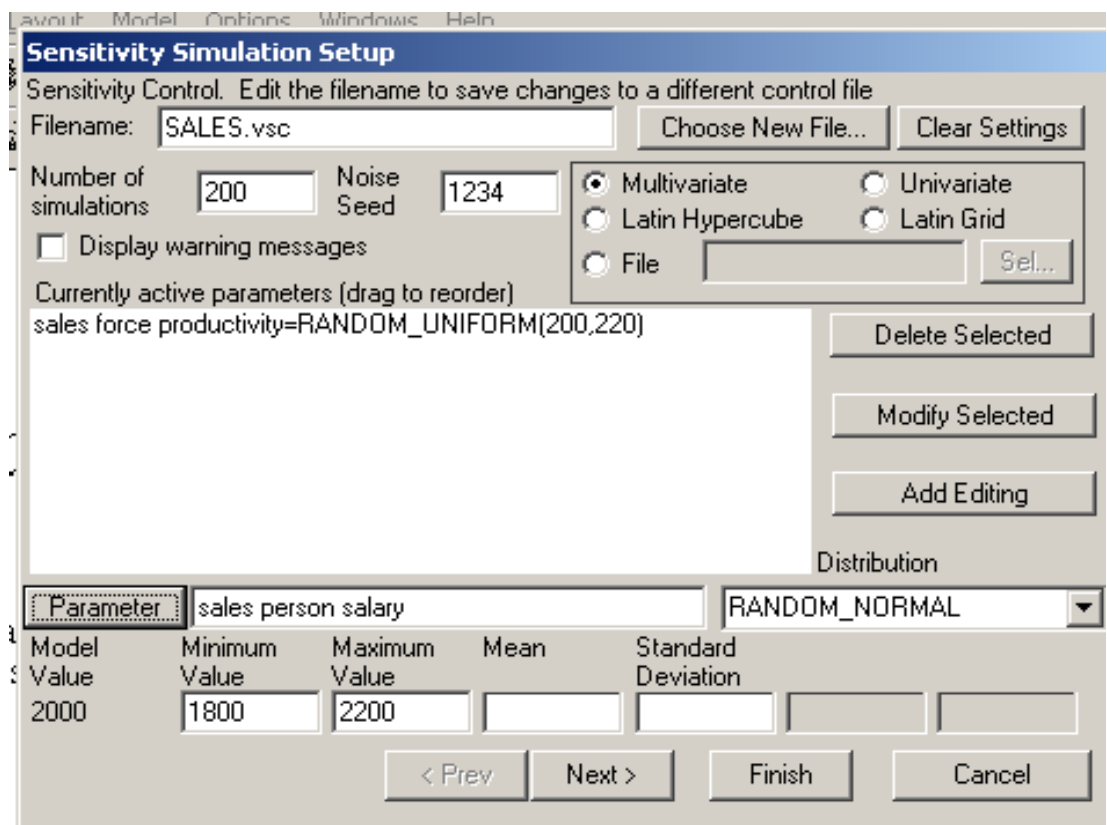


Рисунок 92 – Окно настроек анализа чувствительности

Третья изменяемая константа (переменная) – *sale force adjustment time*. Диапазон ее изменения примем от 3 до 12 при равномерном распределении. В целом окно мастера чувствительности должно выглядеть так (Рисунок 93).

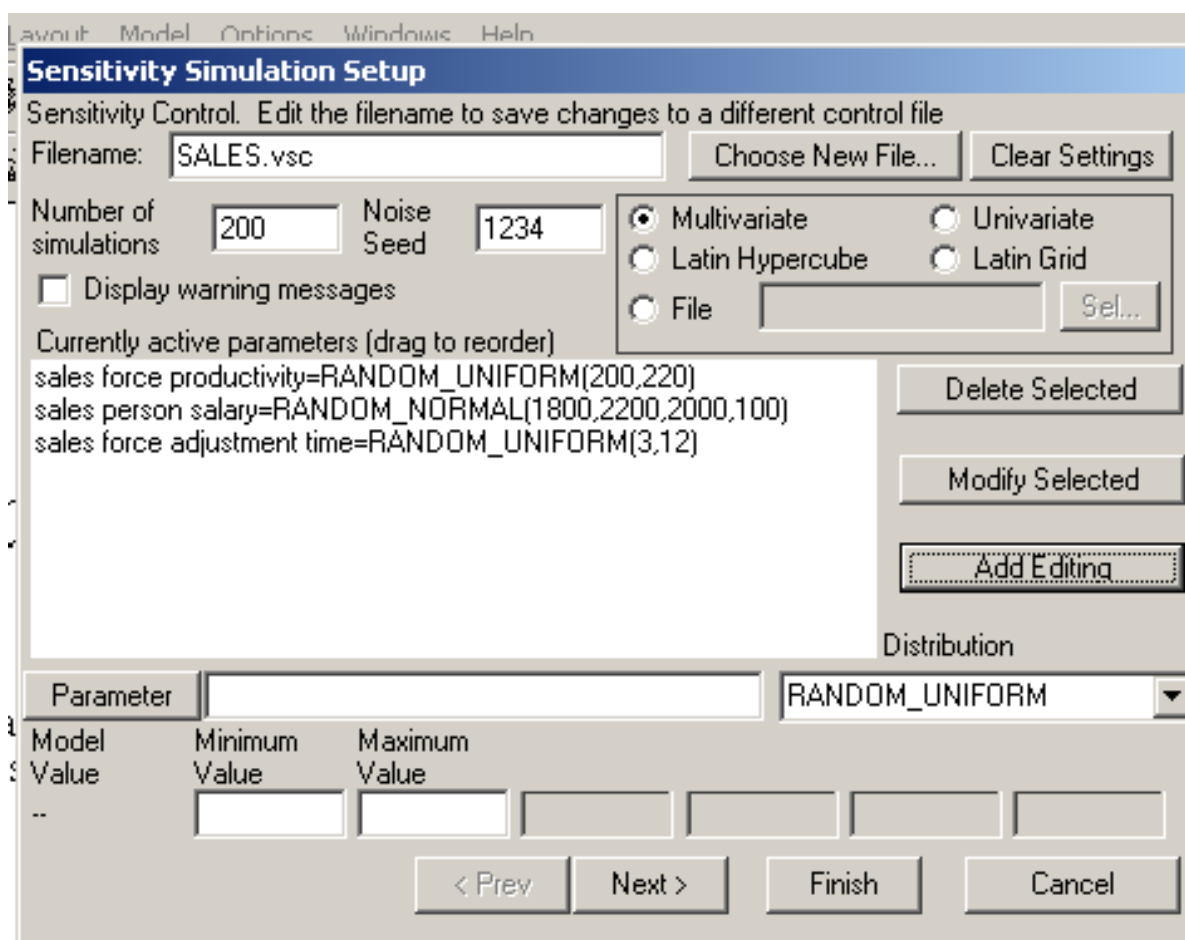


Рисунок 93 – Окно настроек анализа чувствительности

Заполняется поле результативных величин.

После нажатия кнопки **Finish** реализуется алгоритм Монте-Карло. Результаты выдаются в графической форме (Рисунок 94).

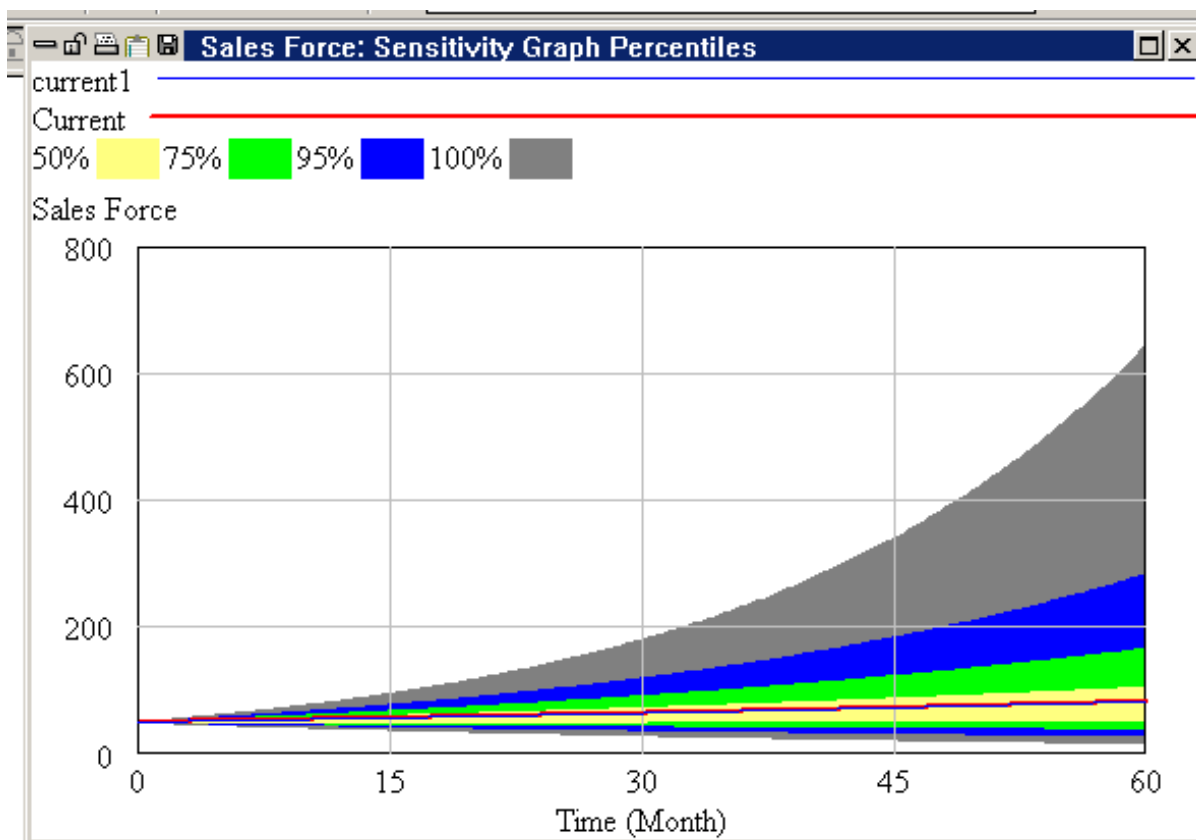


Рисунок 94 – Графическое изображение чувствительности модели

Результаты эксперимента на чувствительность переменной cash flow (Рисунок 95).

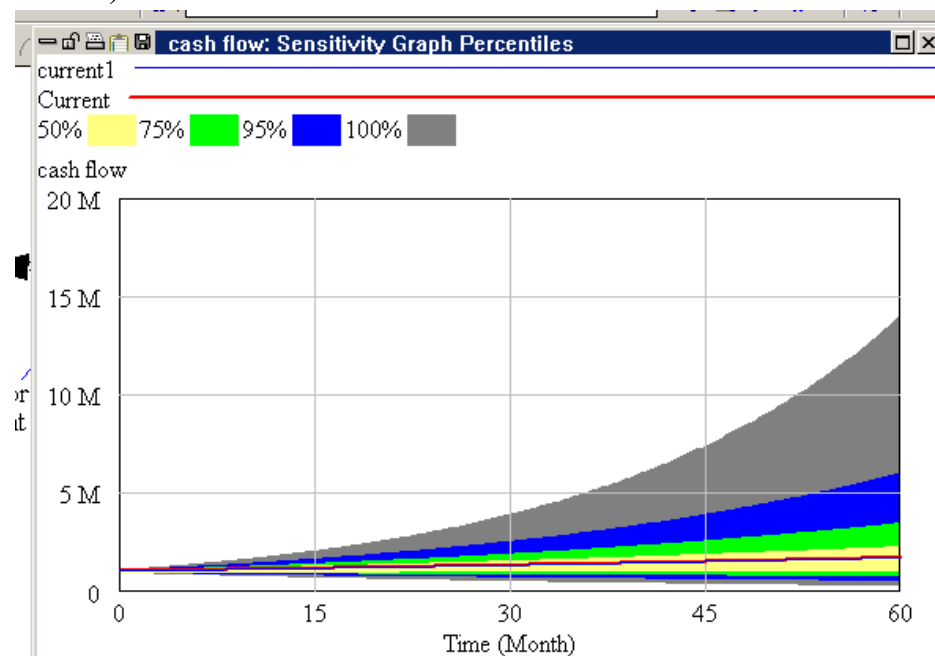


Рисунок 95 – Результаты эксперимента на чувствительность переменной *cash flow*

Графики показывают границы всех значений для Sales Force и cash flow. Представлена чувствительность для вышеназванных трех изменяемых входных параметров. График можно прочесть следующим образом. Линия, обозначенная как *sensitivity*, отражает изменение результирующего признака при первом прогоне модели (с первоначальным, закладываемым в модель, значением константы). Мы видим, что величина потока наличности со временем увеличивается (см. среднюю линию на графике).

Распределение результирующей величины *cash flow*, например, на периоде 45 месяцев с начала моделирования, будет следующим:

Процент появления данного значения величины, %	Границы, млн р.		
	Min/max	45 мес. с начала моделирования	60 мес. с начала моделирования
50	Нижняя	1,0	1,2
	Верхняя	2,0	2,1
75	Нижняя	0,9	0,8
	Верхняя	2,5	3,5
95	Нижняя	0,7	0,6
	Верхняя	4,0	6,0
100	Нижняя	0,6	0,5
	Верхняя	7,0	10,0

В более отдаленной перспективе, через 60 месяцев (в конце моделирования), это распределение будет уже иным. Заметим, что чем больше период времени с начала моделирования, тем больше разброс значений вокруг значения первого прогона, который осуществлялся обычно (но не всегда) с серединой в интервале значений факторного признака. Например, для переменной *sales force productivity*, изменяющейся в диапазоне от 200 до 220 первоначальное значение было равно 210, а для переменной *sale force adjustment time*, изменяющейся в диапазоне от 3 до 12, это значение равно 6.

Если требуется установить разброс результирующего признака более точно, то в Vensim PLE Plus есть инструмент конкретизации графического изображения, когда уменьшается на графике масштаб сетки результирующего признака. Для этого выполняются следующие действия:

- с нажатой клавишей *ctrl* нажать на какой-либо уровень оси Y (например на 5 млн) и удерживая кнопку мыши перетащить курсор к основанию графа (оси X), затем отпустив мышь. Тогда максимальной величиной оси Y станет величина 5 млн, а график станет более точным;
- нажать на кнопку графа чувствительности на панели инструментов (Рисунок 96);

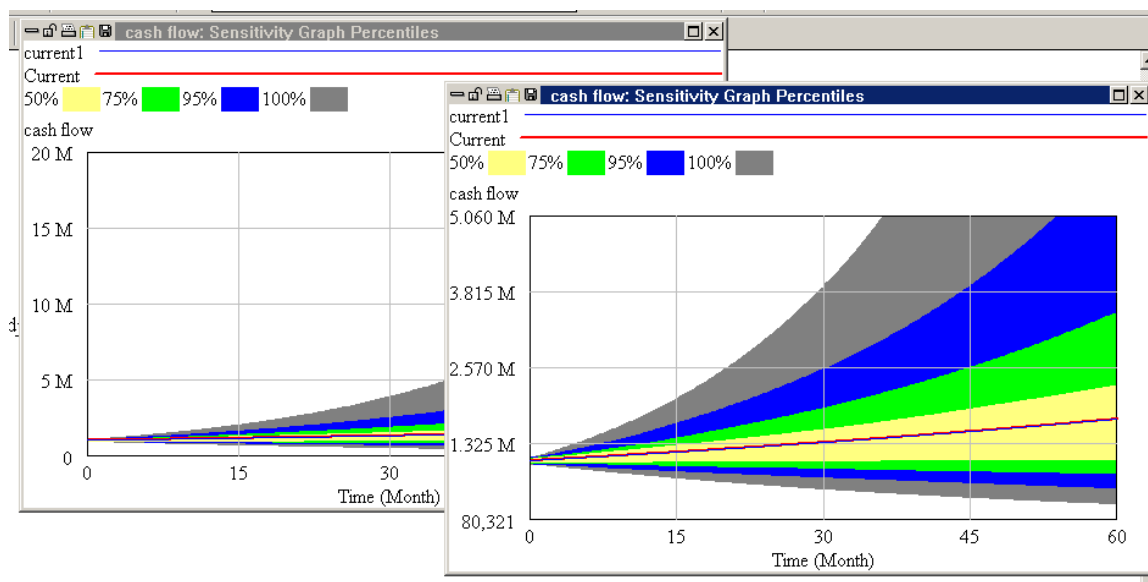



Рисунок 96 – Граф чувствительности переменной

Можно также пронаблюдать непосредственные результаты каждого прогона, используя кнопку  **Sensitivity - Individual Traces** на панели инструментов. Для cash flow результаты 200 прогонов будут такими (Рисунок 97).

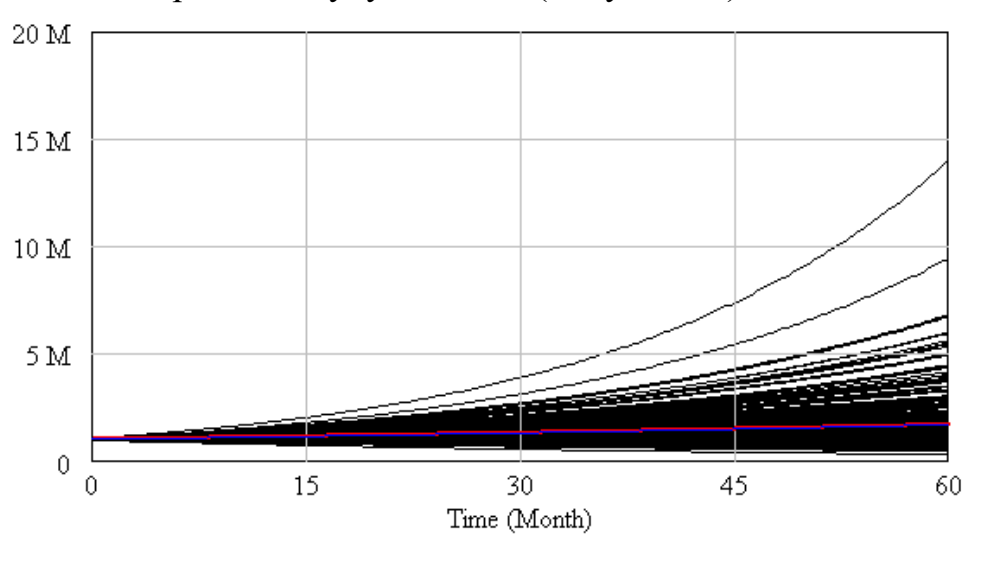


Рисунок 97 – Графическое изображение результатов каждого прогона

3.15. Оптимизационный эксперимент в моделях системной динамики

Моделируя динамику систем на практике бывает необходимым найти оптимальные параметры системы. Например, какая ставка налогов должна быть, чтобы обеспечить максимальные поступления в государственный бюджет. То есть требуется откалибровать элементы модели. Как правило, резуль- тативные характеристики моделей системной динамики чувствительны к изменению входных параметров (констант). Не используя специальный инстру-

ментарий для нахождения оптимальных значений пришлось бы проводить огромное количество экспериментов. В Vensim такой инструментарий есть.

Пример. Компания продает продукцию, имея производственные запасы, которыми нужно управлять, чтобы обеспечить предприятию максимальную прибыль.

Если запасов на складе будет недостаточно, то это приведет к срывам поставок и занижению прибыли. В случае слишком больших запасов влечет за собой увеличение складских расходов и, следовательно, так же снижение прибыли. В качестве результативной переменной в модели выбирается *Прибыль*. В качестве фактора, которым можно варьировать, выбирается минимальная и максимальная величина запасов на складе. Задача решается с использованием оптимизатора, встроенного в программу Vensim (присутствует только в коммерческих версиях). Оптимизатор проводит серию прогонов с изменяющимися значениями факторного признака и выдает графики с результатами первого прогона и прогона с оптимальным значением факторного признака.

3.14. Моделирование систем в АПК

3.14.1. Модель «Зависимость динамики прибыли предприятия от схемы погашения долга»

Постановка задачи

Предприятие занимается производством санитарно-технических изделий. На модернизацию производства требуются инвестиции, для чего планируется взять кредит. Погашение кредита будет производиться из чистой прибыли, являющейся частью балансовой прибыли. Балансовая прибыль формируется как разница между выручкой и общими затратами, определяемыми в данной модели как сумма материальных затрат, затрат на оплату труда и прочих затрат. Предприятие может эффективно управлять только той частью прибыли, которая остается у него после уплаты налогов, т.е. чистой прибылью. Часть чистой прибыли, направляемая на погашение долга, формируется как разница между чистой прибылью и суммой средств, направляемых на расширение производства (переменная *Фонд накопления*) и средств, направляемых в резервный фонд (переменная *Резервный фонд*). Пропорции распределения прибыли между фондом накопления, резервным фондом и средствами, направляемыми на погашение долга, решает руководство предприятия. Оно стремится выбрать такую стратегию развития, чтобы, с одной стороны

уменьшить дебиторскую задолженность, а, с другой стороны увеличить прибыль и размеры производства (часть прибыли, идущая на накопление).

Уровень *Долг* показывает задолженность организации, он формируется как разница между полученным кредитом (темп *Кредит*) и выплатами, производимыми предприятием (темп *Основная выплата*). Размер кредита складывается как сумма *Новых инвестиций*, которые предприятие привлекает в виде очередного кредитного транша, и *Части финансового долга*, оставшегося после очередной, прошедшей в данном месяце, выплаты по долгу. При увеличении долга возрастает риск потерь, что указывает на неспособность организации удерживать свои позиции на рынке. С помощью данной модели можно проследить изменения прибыли при росте (снижении) выручки, увеличении(уменьшении) затрат на производство, а также – как это повлияет на погашение кредита и в какие сроки.

Модель приведена на Рисунок 98.

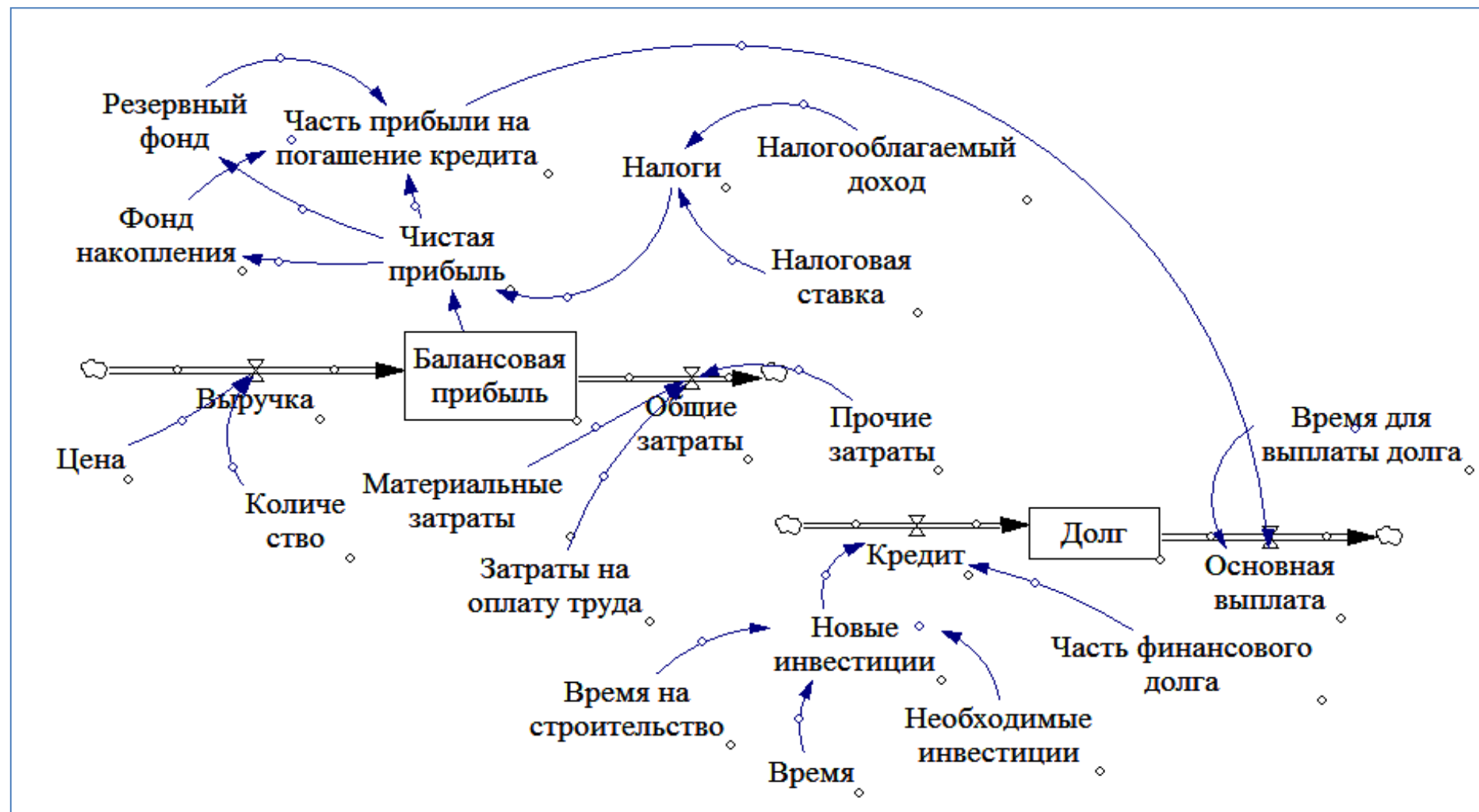


Рисунок 98 – Схема модели

Горизонт моделирования – 5 месяцев. Выражения модели и значения констант приведены в приложении 1.

Один из путей увеличения прибыли – это повышение цены. Данная стратегия возможна, так как предприятие является монополистом. Модель может показать, как изменится динамика погашения долга при увеличении цены продукции с 500 до 700 рублей.

Результаты представлены на Рисунке 99.

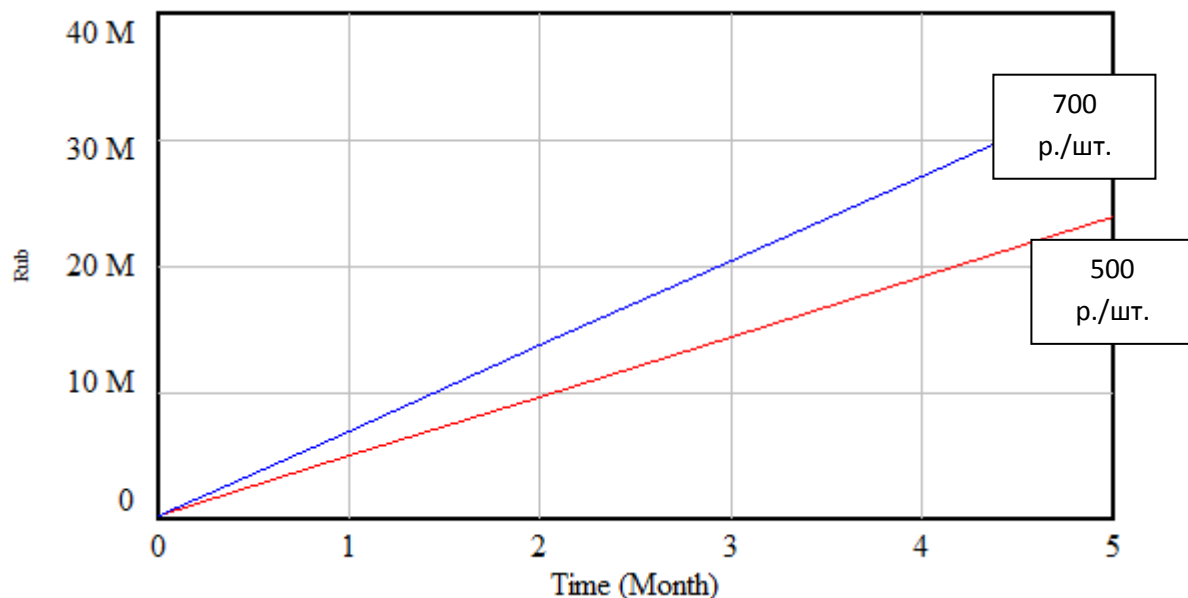


Рисунок 99 – Динамика балансовой прибыли при ценах 500 и 700 руб./шт.

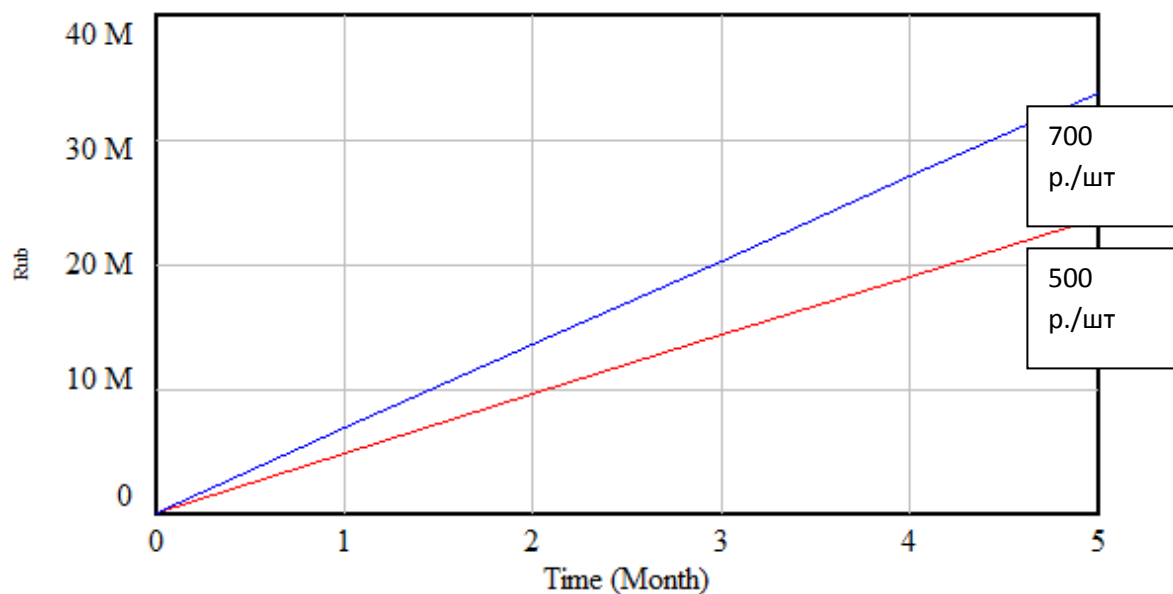


Рисунок 100 – Динамика чистой прибыли при ценах 500 и 700 руб./шт.

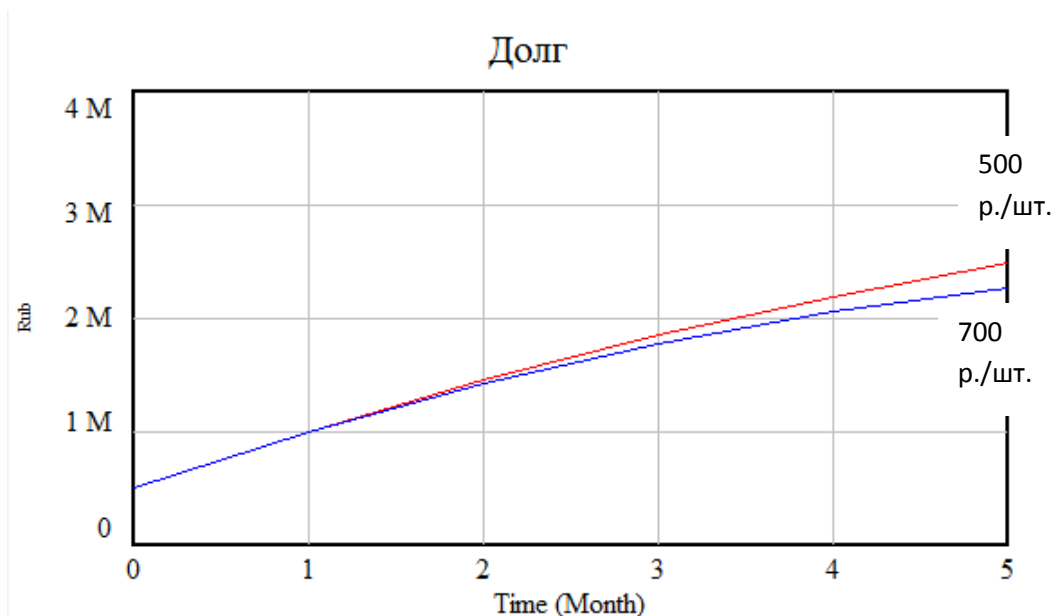


Рисунок 101 – Динамика долга предприятия при ценах 500 и 700 руб./шт.

На данной модели можно также посмотреть, как изменит состояние предприятия (его балансовую и чистую прибыль, а также задолженность) изменение таких показателей как: количество выпускаемой продукции, часть прибыли, направляемая на накопление, размер резервного фонда, налоги, время строительства, величина материальных затрат различных видов.

3.14.2. Модель «Выбор оптимальной стратегии использования ресурсов»

Постановка задачи

Необходимо смоделировать деятельность автомобильного завода для нахождения оптимальной производственной стратегии.

Автомобильный завод производит два вида автомобилей: легковой (А) и грузовой (В). Для производства каждого типа автомобиля требуется определенное количество единиц металла. Качество металла представлено тремя видами. От качества металла зависит количество металла, необходимого для производства каждого вида автомобиля. Продажи автомобилей определяются производимым количеством и коэффициентом продаж, который зависит от стоимости каждого типа автомобиля – с увеличением стоимости автомобиля коэффициент продаж уменьшается (так реализуется закон спроса). Выручка завода зависит от количества проданных автомобилей и их цены. При производстве автомобилей завод несет следующие затраты: на производство, хранение, транспортировку и реализацию в автосалонах автомобилей, а также – оплату труда и амортизацию оборудования.

Схема модели приведена на Рисунке 102.

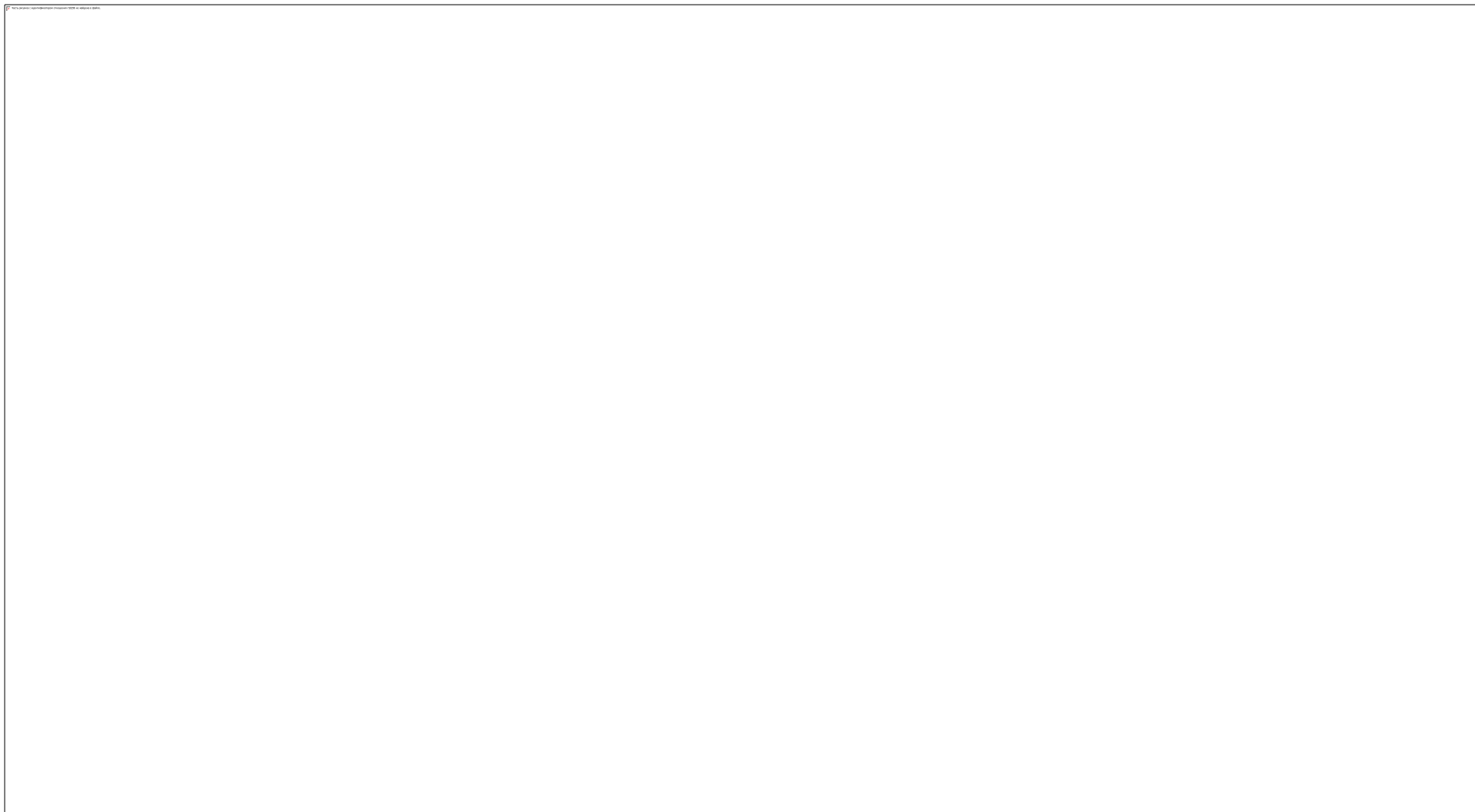


Рисунок 102 – Схема модели «Выбор оптимальной стратегии
использования ресурсов»

Значения констант, связи между элементами модели и их типы определены в **Приложении 2**.

Данную модель можно использовать, например, для выбора металла, из которого сделаны машины оптимального качества, то есть такого, при использовании которого прибыль производителя является максимальной.

При использовании металла класса качества №1 чистая прибыль имеет следующую динамику (Рисунок 103).

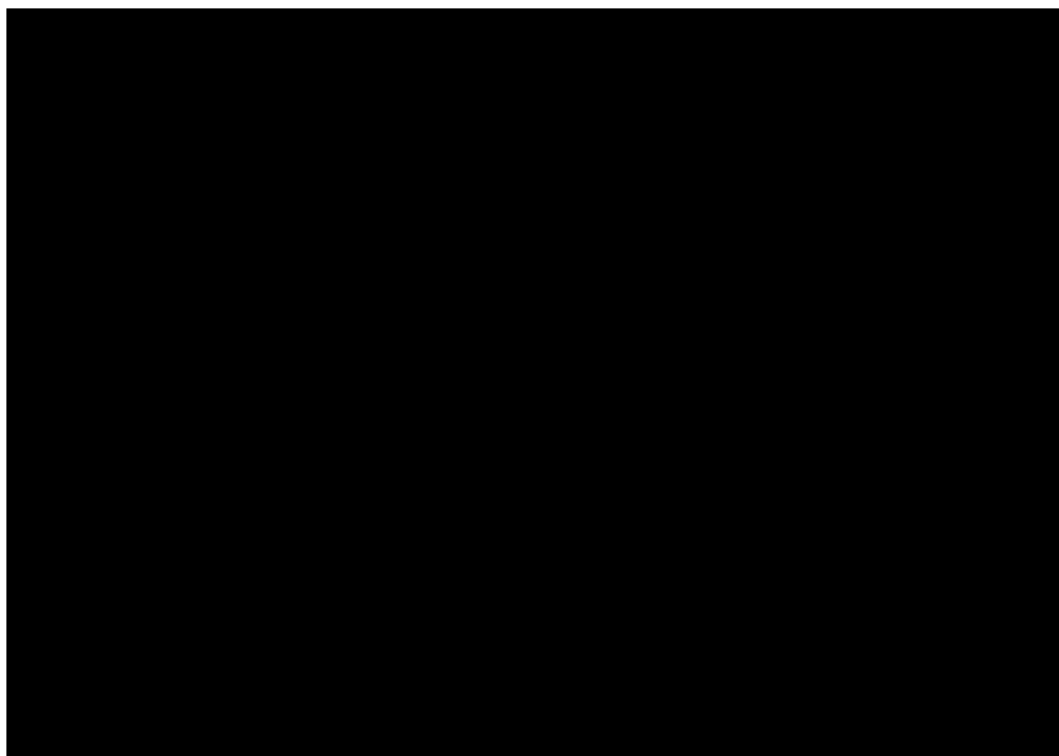


Рисунок 103 – Чистая прибыль при классе качестве металла = 1.

При использовании металла класса качества № 2 прибыль убывает, так как это более дорогой металл (цена в 2 раза выше, нежели металла класса качества № 1), что сказывается на спросе и предприятие теряет возможную прибыль за чет этого (Рисунок 104).

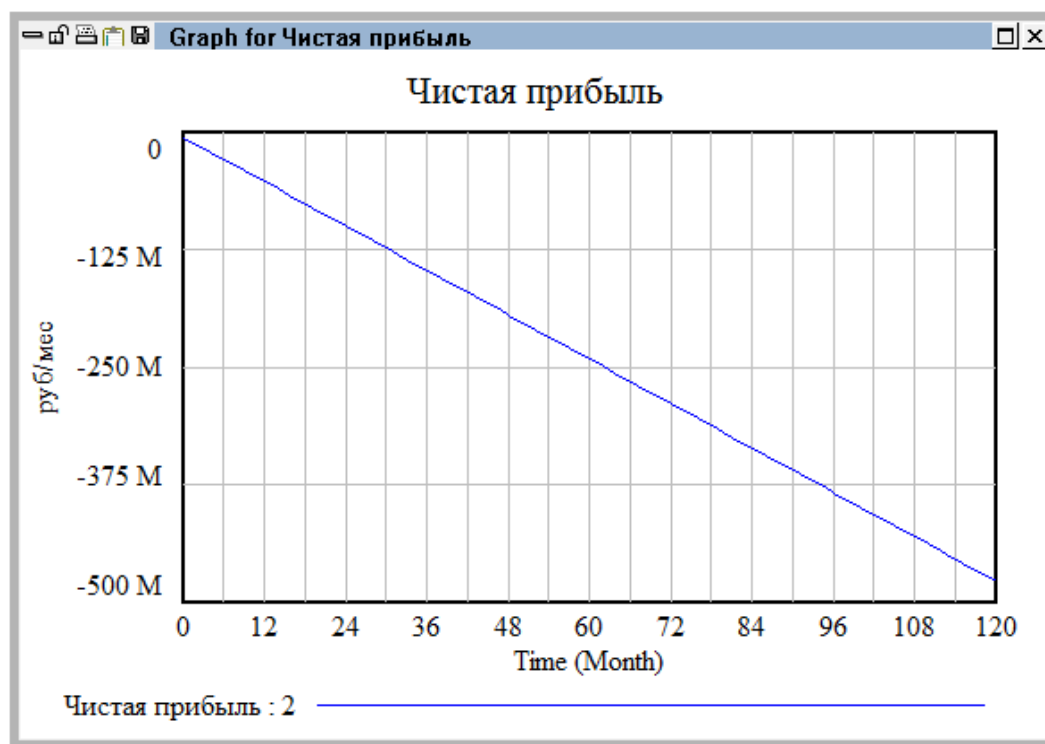


Рисунок 104 – Чистая прибыль при классе качестве металла = 2.

При использовании металла класса качества № 3, цена автомобиля также вырастет, но незначительно по сравнению с применением металла класса качества № 2, так как при общем падении спроса предприятие получает больше выручки за счет более высокой цены автомобиля, что выражается в росте прибыли (Рисунок 105).

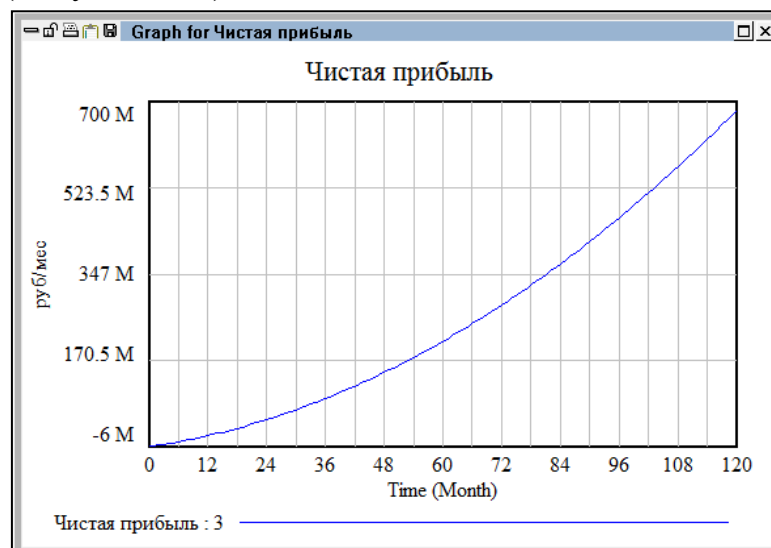


Рисунок 105 – Чистая прибыль при классе качества металла № 3.

При сравнении данных стратегий (три графика помещены на одной координатной плоскости) (Рисунок 106) видно, что наиболее выгодной является третья производственная стратегия – использование металла класса качества № 3.

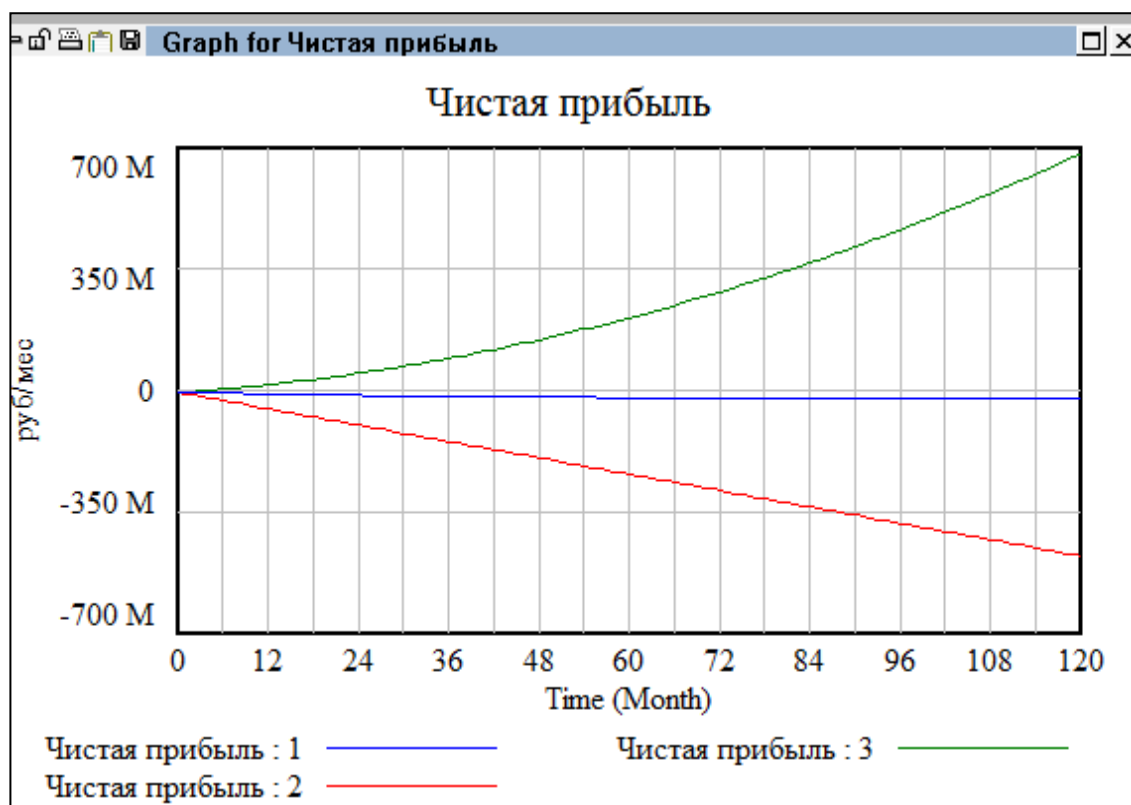


Рисунок 106 – Графики чистой прибыли разных стратегиях.

По данной модели также можно проследить перспективу развития предприятия при изменении следующих условий:

Сумма амортизации (которая является производной от стоимости основных средств);

затраты на хранение автомобиля;

коммерческая надбавка, то есть норматив рентабельности, закладываемый предприятием в цену продукции;

расходы на продажу в автосалонах;

величина оплаты труда;

цена автомобиля;

численность сотрудников.

3.14.3. Модель «Стратегии кормопроизводства сельскохозяйственного предприятия»

Постановка задачи

Жирность молока определяет его качеством и, следовательно, она определяет закупочную цену молока и выручку предприятия. Жирность молока определяется в свою очередь качеством корма, которое в данной модели для упрощения представлено одним из множества показателем – содержание в корме переваримого протеина. Корм более высокого качества имеет более

высокую себестоимость, так как для его производства используется более совершенная и дорогая технология. Поэтому в модель введено следующее условие: если это – корм первого сорта, то он дороже, в расчете на одну голову в месяц это составляет 4800 руб. (*Качество корма* = 1 (первого сорта)). Если же используется корм более низкого, второго сорта (*Качество кормов* = 2), то себестоимость его производства в количестве, равном среднемесячной потребности одной головы скота составляет 3700 руб. На заготовку более дешевого несортного корма предприятие затрачивает 2800 руб./месяц/гол.

Главная цель работы предприятия – повышение прибыли и рентабельности. Дешевый корм производить выгоднее, однако кормоотдача от него ниже, что дает более низкую выручку от продажи молока.

Качество корма определяет также величину надоев на одну голову. Если скоту скармливается корм первого сорта, то надой на одну голову в месяц равен 300 л.

Определенная часть поголовья коров погибает вследствие болезней (падеж скота). После окончания активной фазы лактации коровы подлежат забою. Таки образом, хозяйство продает мясо КРС.

Прибыль формируется как разница между выручкой и затратами на производство. Кроме того, при ее определении вычитается убыток, понесенный от падежа скота. Величина убытка определяется произведением количества голов падежа на стоимость (цену) одной головы, которая бы была получена при реализации скота на мясо.

Летом себестоимость корма понижается, так как отсутствуют затраты на отопление и снижены затраты на освещение, а также в рацион вводится зеленый корм. Кроме того, это происходит вследствие роста продуктивности скота в весенне-летний период (Рисунок 107).

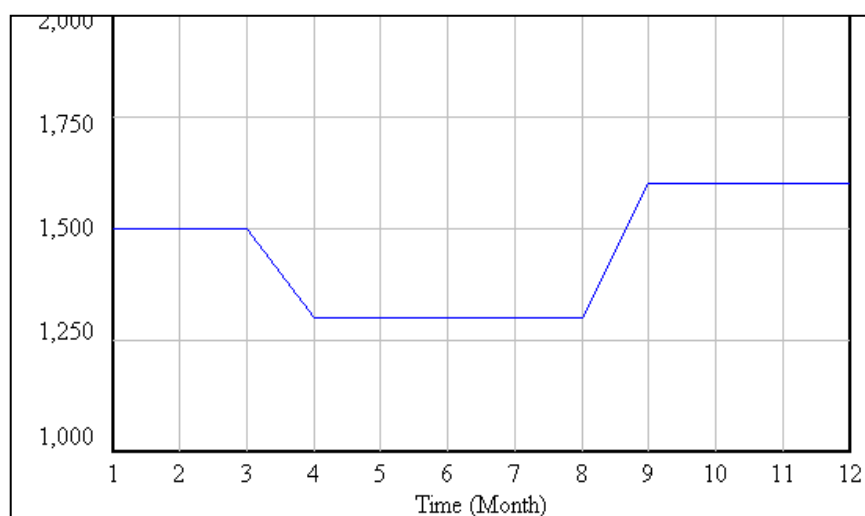


Рисунок 107 – Себестоимость содержания одной головы скота

Себестоимость производства молока в данной модели разделена на две составляющие: себестоимость корма и себестоимость содержания скота. Прибыль предприятия от производства молока складывается как разница между выручкой и себестоимостью продукции зависит от падежа скота. Выручка определяется как сумма произведения численности скота на надой от одной головы и цену литра молока и выручкой от продажи забитого скота. поголовье коров пополняется за счет собственного воспроизводства.

Среди выражений модели есть специфические. Например, стоимость кормов на одну голову скота в месяц определяется следующим образом:

Стоимость кормов на одну голову скота в месяц =
IF THEN ELSE(Качество кормов =1, 4800 , (IF THEN ELSE(Качество
кормов =2, 3700 , 2800)))
*Units: руб/(Month*гол.).*

Это означает, что при качестве корма, равном единице, стоимость корма составляет 4800 руб./мес./гол., если качество корма равно 2, то данный показатель составляет 3700 руб./мес./гол. При других условиях стоимость корма составляет 2800 руб./мес./гол.

Схема модели приведена на Рисунок 89.

Выражения, используемые при разработке модели приведены в Приложении 3. Результаты прогонов модели представлены на Рисунке 108. На нем показано, что было проведено два погона модели. Первый с качеством корма, равным 1 (более качественный, но дорогой корм), второй – с низкокачественным, но дешевым кормом.

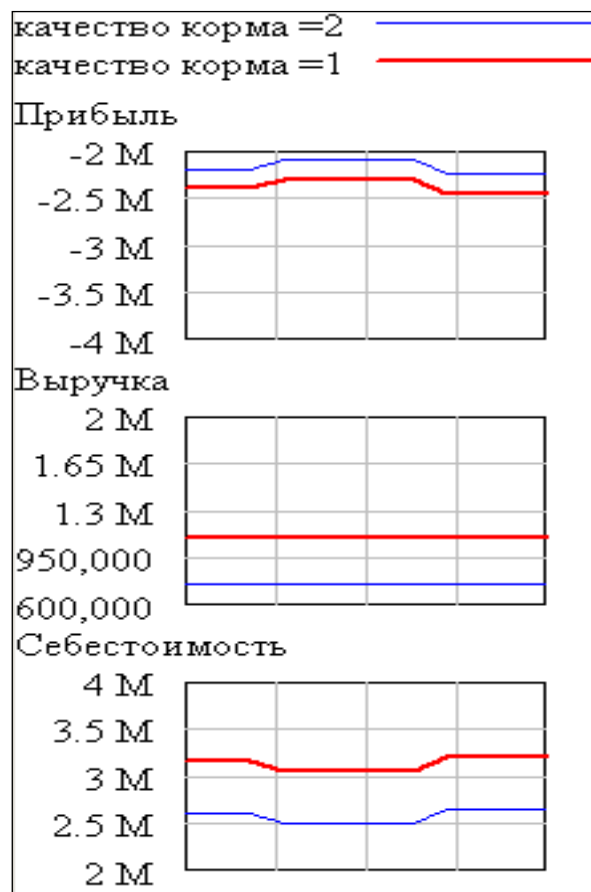


Рисунок 108 – Причинно-следственный граф переменной «Прибыль»

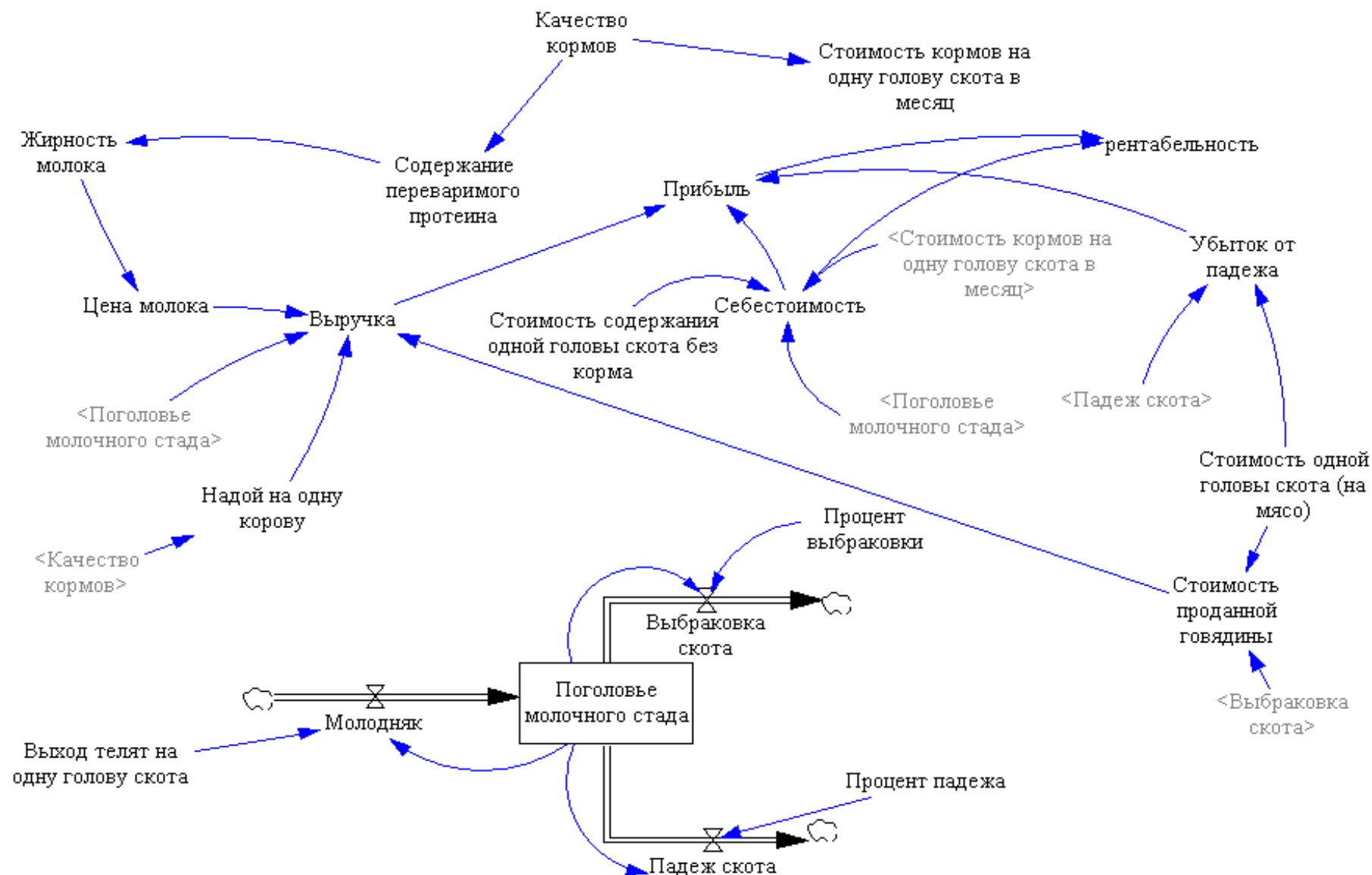


Рисунок 109 – Схема модели

По результатам прогонов видно, что при производстве и скармливании скоту корма первого сорта (более высокого качества) прибыль предприятия меньше. Это происходит потому, что производство корма более высокого качества наиболее затратно (см. «Себестоимость»), что не окупается адекватным количеством корма. Графики также показывают, что в летнее время себестоимость производства корма снижается. При неизменной цене это приводит к росту прибыли. То есть наиболее выгодной стратегией является производство и использование корма более низкого качества, но являющегося наиболее дешевым.

На данной модели также можно оценить влияние на результаты работы предприятия таких факторов, как процент падежа, выход телят на одну голову, стоимость содержания одной головы.

3.14.4. Модель «Динамика распространения продукции предприятия на рынке региона»

Постановка задачи

Предприятие производит посуду из нержавеющей стали. Для планирования производства предприятия производит маркетинговые исследования – планирует объемы возможных будущих продаж. Часть населения города (рынка) – 1000 человек – уже пользуется купленной у данного предприятия посудой (*Купившие товар*). Остальная часть города является потенциальными покупателями (*Потенциальные покупатели*). Их численность составляет 1 млн человек. И те и другие образуют общий рынок города (*Общий рынок*).

Купившие посуду контактируют (рассказывают о качестве посуды) как друг с другом, так и с потенциальными покупателями данного предприятия. Согласно анкетным данным предприятия каждый купивший посуду в среднем рассказывает о ней своим знакомым 20 раз в месяц (*Общаяемость*), которая измеряется в количестве контактов на одного человека в месяц. Общее количество контактов будет находиться как произведение *Общаяемости* на *Количество купивших товар*. Так как наиболее действенным видом рекламы является реклама при личных контактах, то предприятие интересуется не общее количество контактов купивших посуду, а только их контакты с потенциальными покупателями. Доля потенциальных покупателей определяется как отношение численности потенциальных покупателей к общему рынку (*Потенциальные покупатели/Общий рынок*). Соответственно, контакты потенциальных покупателей с купившими товар будет определяться как произведение контактов купивших товар и доли потенциальных покупателей в общем рынке (*Контакты купивших товар*доля потенциальных покупателей в общем рынке*). Данная величина измеряется в контактах в месяц.

То, какая часть потенциальных клиентов покупает посуду, зависит от степени убеждаемости потенциальных клиентов (*Убеждаемость*). Согласно социологическим опросам только один из ста человек, с которым купившие «поделились» информацией о посуде, решает ее приобрести (0,01 человек/контакт). Количество решивших купить товар в месяц определяется как *Контакты потенциальных покупателей с купившими товар*Убеждаемость*.

Количество новых покупателей из числа потенциальных покупателей будет равно *Новым покупателям*.

Все, отмеченное выше взаимосвязано следующим образом:

Общаяемость = 20 (контактов/человека/месяц).

Купившие товар = 1000 + *Новые покупатели* (человек).

Потенциальные покупатели = 1 млн (человек).

Количество решивших купить товар (в месяц) = *потенциальных покупателей с купившими товар*Убеждаемость* (человек/месяц).

Доля потенциальных покупателей в общем рынке = *Потенциальные покупатели/Общий рынок* (доля).

Убеждаемость = 0,01 (человек/контакт).

Горизонт моделирования составил 100 месяцев. Схема модели приведена на Рисунке 110.

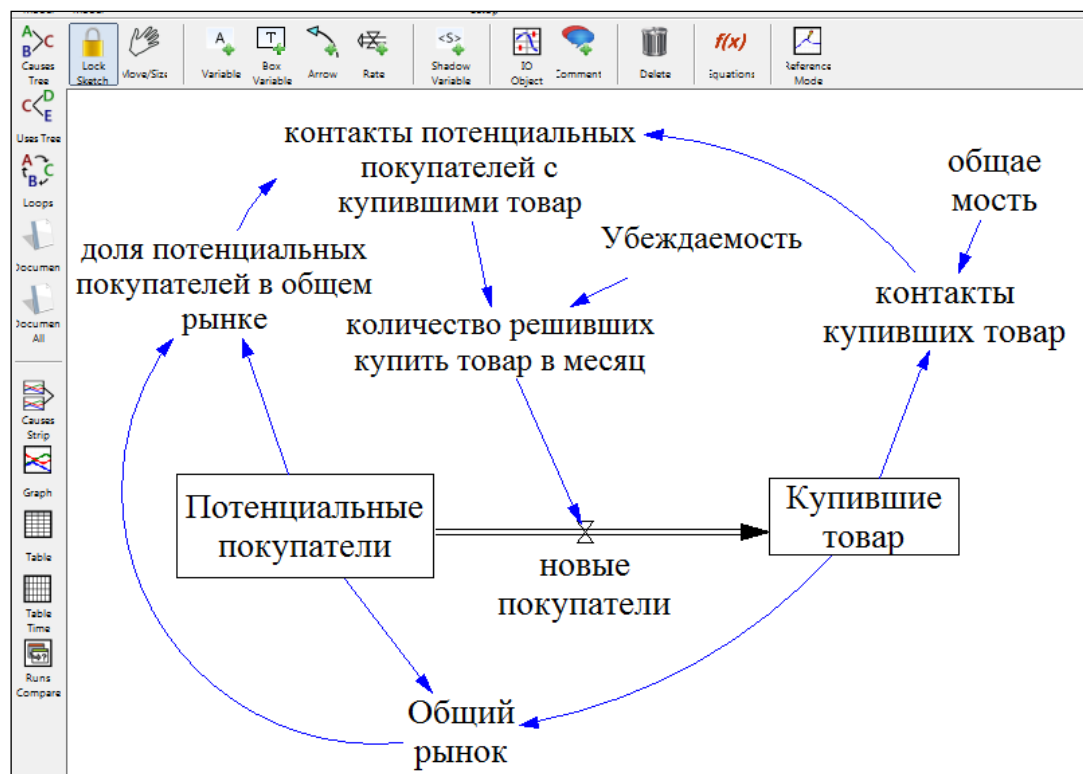


Рисунок 110 – Схема модели «Динамика распространения продукции предприятия на рынке региона»

По данной модели можно прогнозировать объем продаж предприятия. Здесь также можно «проиграть» ситуацию с продажами в случае, если изменится общаемость владельцев посуды или убеждаемость новых клиентов. На Рисунке 111 представлены результаты прогонов моделей с различными величинами убеждаемости. В исходной модели она составляла 0,01 человек/контакт. Если она по каким-либо причинам возрастет до 0,05, то есть каждые пятеро из ста решат покупать посуду, то предприятие быстрее завоюет рынок данного города.

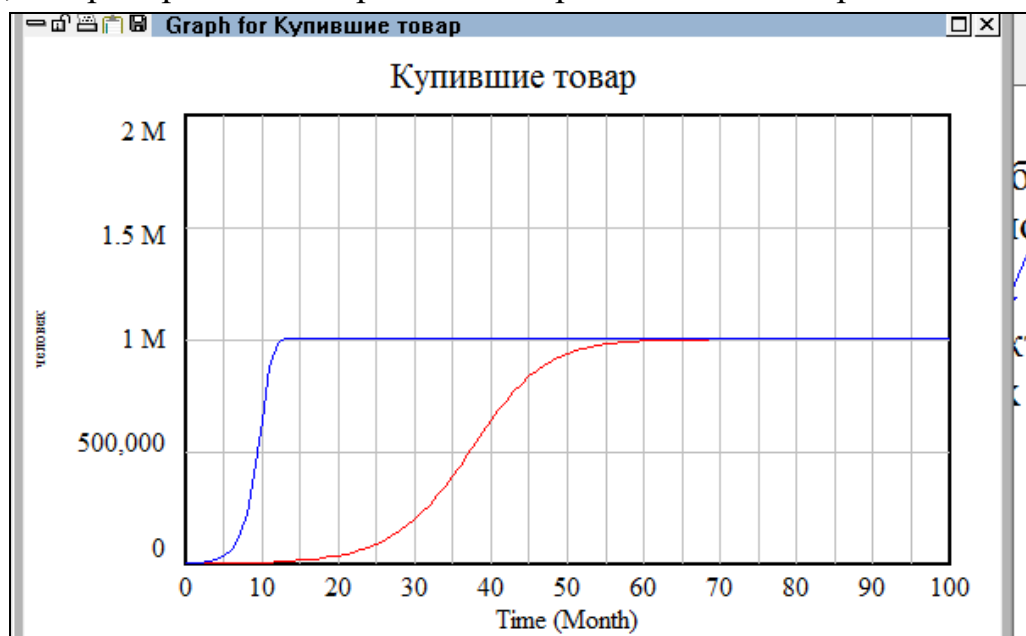


Рисунок 111 – Результаты прогона модели по переменной «Купившие товар»

Более крутая линия – при убеждаемости 0,05 человек/контакт. Как видно при этом предприятие полностью завоюет рынок за 12 месяцев (при прежнем значении – за 60 месяцев).

3.14.6. Модель «Оценка вариантов инвестиционного проекта строительства цеха по переработке молока сельскохозяйственным предприятием»

Постановка задачи.

Сельскохозяйственное предприятие планирует построить цех по переработке молока, так как на рынке переработанное и упакованное молоко стоит в 2-3 раза дороже. При строительстве цеха планируется использовать уже имеющееся на предприятии помещение, поэтому капиталовложения планируется направить на покупку нового оборудования стоимостью 200 тыс долл. Строительство планируется осуществлять в основном за счет заемных средств (60 %), остальные средства – собственные. Чем больше предприятие привлечет заемных средств, тем меньше сроки строительства, и тем раньше предприятие сможет выйти на рынок со своей продукцией. Однако, при этом велика величина процентов за поль-

зование заемными средствами. С другой стороны, при небольшом объеме заемных средств потребуются затратить собственные средства, изъяв их из оборота, что негативно скажется на основном производстве. Таким образом, необходимо найти оптимальный вариант между сроками строительства и размером привлекаемых средств.

Банк может предоставить кредит на срок до трех лет.

Амортизационные отчисления определяются на основании срока полезного использования приобретенного оборудования (10 лет).

Процесс реализации и расчетов за продукцию включает следующие стадии:

- отгрузка произведенной продукции,
- выставление счетов покупателям,
- ожидание оплаты за продукцию (дебиторская задолженность),
- поступление денег на расчетный счет.

При этом учитывается, что, как правило, какая-то часть дебиторской задолженности не будет оплачена вовремя. Доля потерь вследствие этого планируется на уровне 0,06.

Доля постоянных издержек определена в размере 60 % общих издержек предприятия. При определении финансового результата по итогам отчетного периода наряду с издержками производства учитываются налоговые выплаты (обобщенно приняты в размере 40 %).

Так как в максимизации прибыли предприятия играют роль, с одной стороны, сроки строительства, а с другой – величина заемных средств, то в имитационной модели рассматривается два варианта финансовой стратегии предприятия:

Стратегия № 1 «Значительная величина заемных средств (короткие сроки строительства цеха)».

Стратегия № 2 «Незначительная величина заемных средств (более длинные сроки строительства)».

Имитационная модель включает четыре «уровня»: «продукция, ожидающая реализации», «дебиторская задолженность», «долг» и «основной капитал».

Прогнозирование осуществляется на период 3 года.

Шаг моделирования – 0,015 года.

Уравнения модели приведены в **приложении 4**.

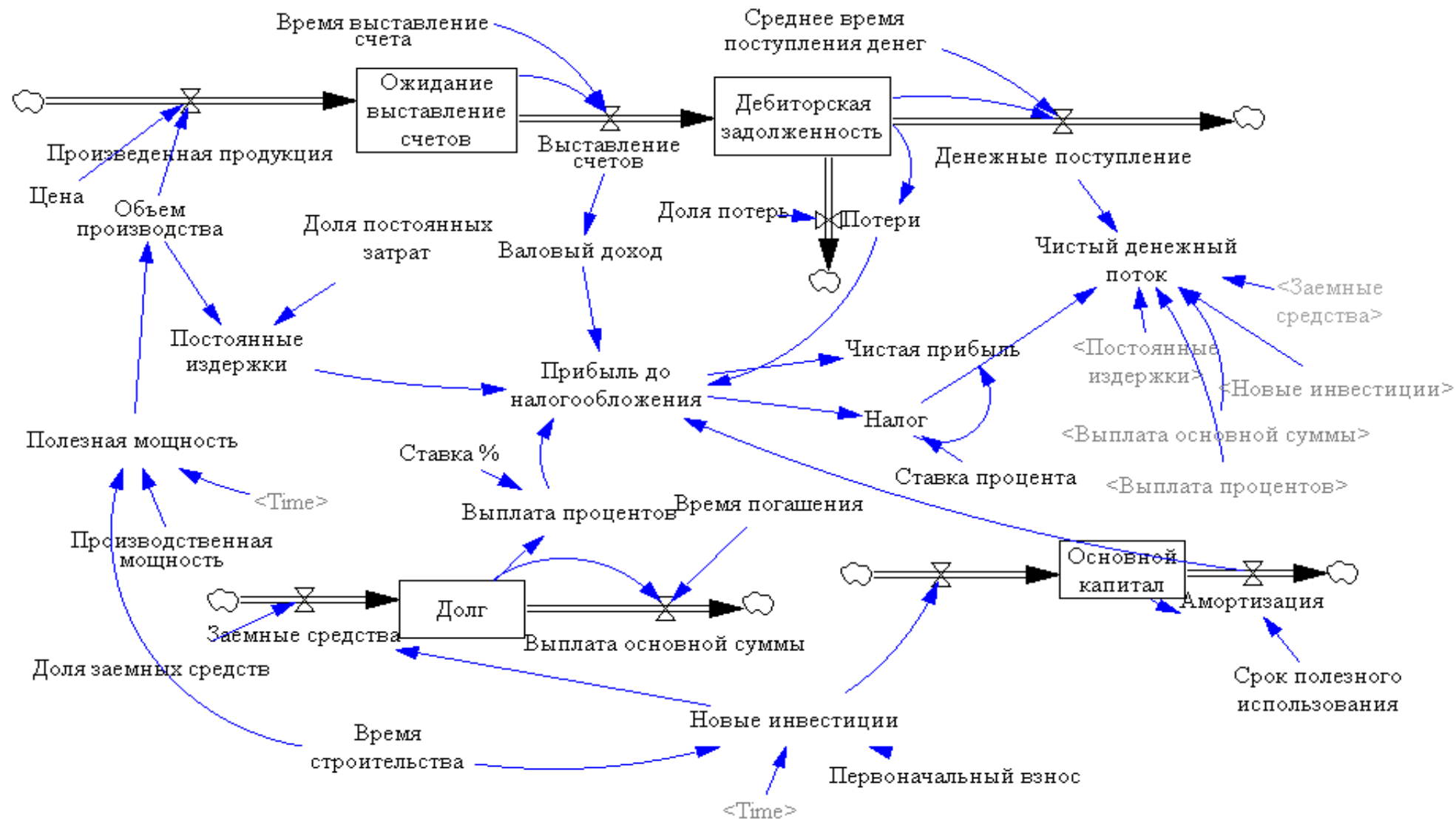


Рисунок 112 – Схема модели

Результаты прогона и сравнительная оценка стратегий развития

Целью направленного вычислительного эксперимента в данной модели – прогнозирование прибыли при различных сроках строительства цеха по переработке молока, обусловленными различными объемами заемных средств.

Стратегия № 1 – «Минимально короткие сроки строительства и ввода в эксплуатацию цеха по переработке молока». Время строительства – 0,5 года.

Стратегия № 2 – «Увеличенные сроки строительства, связанные с минимальным привлечением заемных средств». Время строительства – 2 года.

Итоговым показателем деятельности любого предприятия является прибыль. Динамика прибыли при сравниваемых вариантах (графический (Рисунок 113), табличный (Рисунок 114) и причинно-следственный (Рисунок 115) графы результатов моделирования).

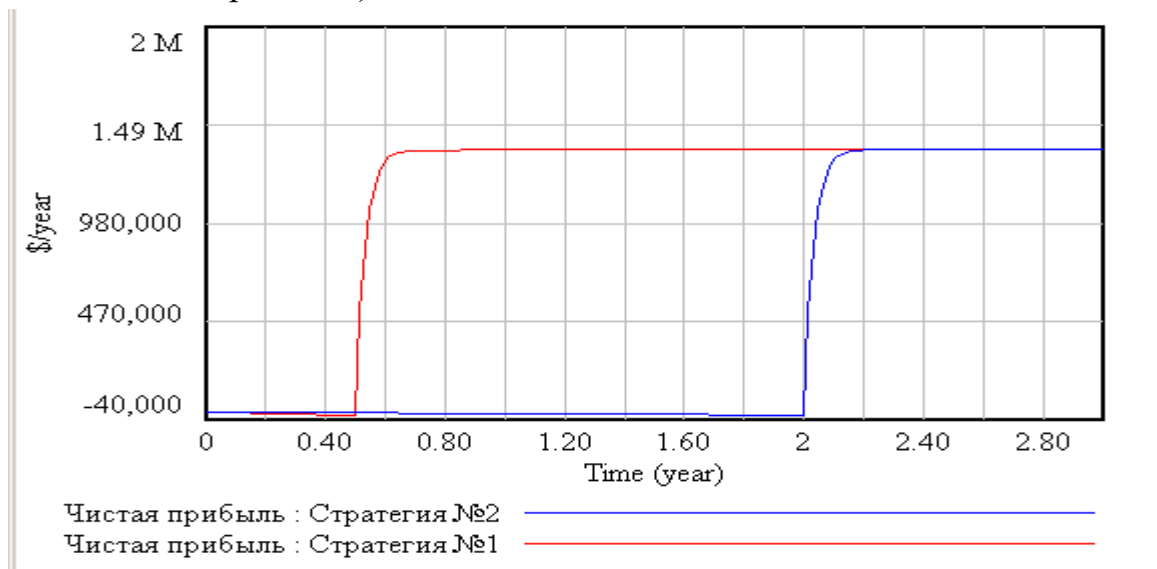


Рисунок 113 – Динамики изменения чистой прибыли при различных стратегиях развития

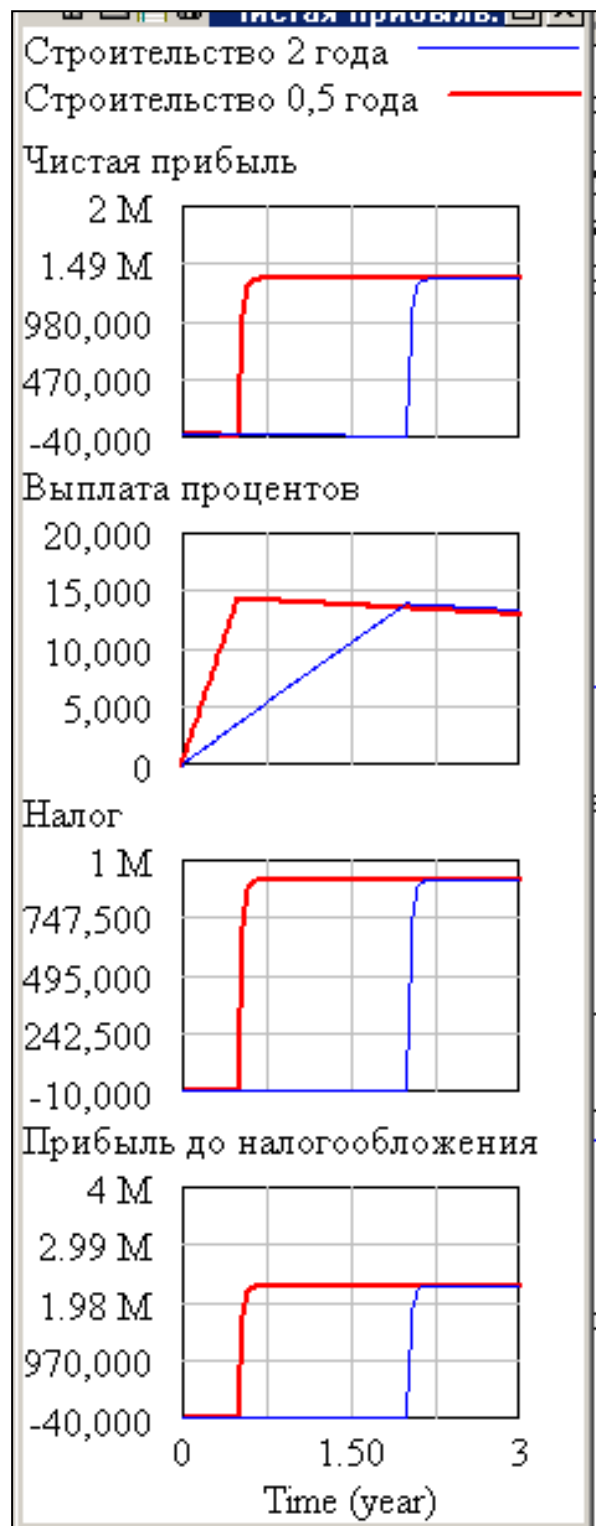


Рисунок 114 – Причинно-следственный граф элемента модели «Чистая прибыль»

Time (year)	"Чистая прибыль"	Чистая прибыль
0	0	0
0.015625	Runs:	-161.25 -645
0.5	Стратегия №2	-4922.84 -20771.4
1	Стратегия №1	-9395.84 1.35961e+006
1.5		-13473.2 1.36117e+006
2		-18281.5 1.36256e+006
2.5		1.36181e+006 1.3638e+006
3		1.36311e+006 1.3649e+006

Рисунок 115 – Таблица значений чистой прибыли

Из Рисунков 113-115 видно, что срок строительства определяется, прежде всего, размером привлекаемых средств. Чем короче время строительства, тем раньше предприятие начнет получать прибыль от животноводства. Но на начальном этапе при коротких сроках строительства требуется значительное вложение средств и, если эти средства – заемные, то величина процентов по займу будет высока. Динамика долга предприятия приведена на Рисунок 116. А причинно-следственный граф элемента модели «долг» приведен на Рисунок 117.

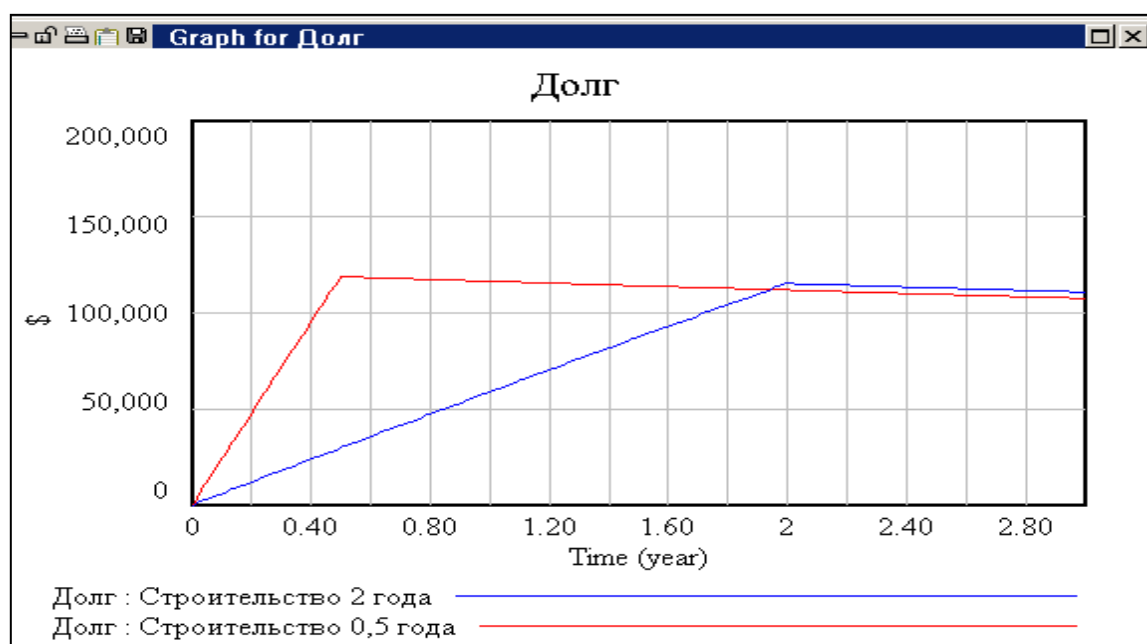


Рисунок 116 – Размеры долга предприятия при разных сроках строительства

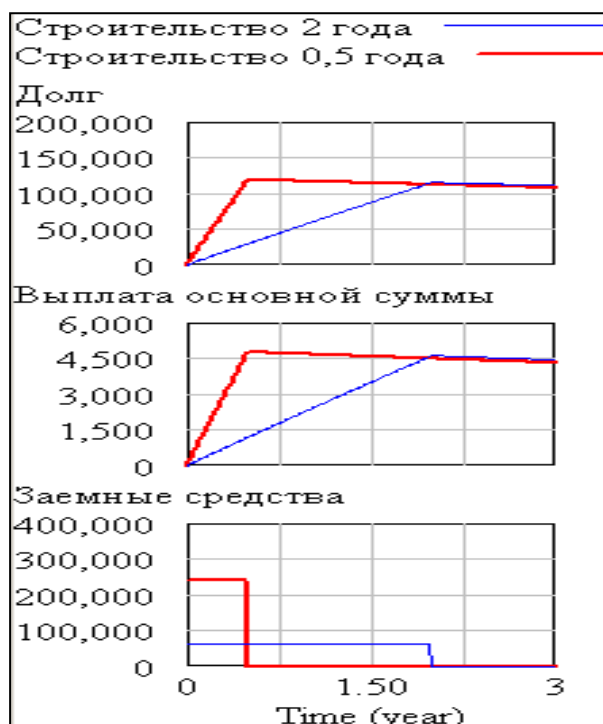


Рисунок 117 – Причинно-следственный граф элемента модели «долг»

Из Рисунок 116 и 117 видно, величина кредитов при малых сроках строительства значительно возрастает. Сельскохозяйственные предприятия из-за недостатка свободных денежных средств вынуждены использовать, как правило, стратегию №2 (увеличенные сроки строительства). Однако, при возможности получения займов больших размеров возможна и вторая стратегия.

3.14.6. Модель «Прогнозирование объемов валовой продукции сельского хозяйства региона и доходов населения»

Постановка задачи

Необходимо разработать модель формирования валовой продукции сельского хозяйства региона. В основу взята модель, разработанная в ГУУ [6].

. От стоимости произведенной продукции зависит заработная плата работников и доходы населения. Стоимость произведенной валовой продукции зависит в данной модели от следующих факторов – квалификация (*Коэффициент эластичности трудовых ресурсов*) и численность рабочей силы (*Численность занятых*), научно-технический прогресс. Последний фактор выражен через стоимость основных фондов (переменная *Основные фонды*) и отдачу от них (*Коэффициент эластичности основных фондов*, *Темп научно-технического прогресса*).

При моделировании влияния НТП и квалификации работников использована широко известная неоклассическая модель экономического роста Кобба-Дугласа [7]. Данная макроэкономическая модель позволяет оценить вклад различ-

ных факторов производства в увеличении объема производства или национального дохода. Эта функция имеет следующий вид:

$$Y = AK^{\alpha}L^{\beta}$$

где Y – объем производства, K – капитал, L – труд, A , α , β – параметры или коэффициенты производственной функции: A – коэффициент пропорциональности; α и β – коэффициенты эластичности объема производства по затратам труда и капитала.

Для расчета частных коэффициентов эластичности, которые в общем виде можно представить в виде $E(x_i)$, использовано корреляционно-регрессионное уравнение. Коэффициент эластичности показывает, на сколько процентов в среднем изменится значение результативного признака при изменении факторного на 1 % при постоянном (фиксированном) уровне других факторов:

$$E_{x_i} = a_i \frac{\bar{x}_i}{\bar{Y}},$$

где a_i – частный коэффициент уравнения регрессии; \bar{x}_i – среднее значение факторного признака; \bar{Y} – среднее значение результативного признака.

Для расчета корреляционно-регрессионного уравнения используем данные по региону, где имеется 11 сельскохозяйственных предприятий (табл. 7).

Таблица 7 – Статистические данные по сельскохозяйственным организациям района

№	Стоимость основных фондов на 100 га пашни, тыс.р. X1	Численность работников на 100 га пашни, чел. X2	Выход валовой продукции на 100 га с.-х. угодий, тыс р. Y
1	949,8	3,7	586,4
2	919,4	1,0	179,8
3	1767,3	1,7	82,4
4	1448,7	3,2	267,8
5	1957,1	3,2	94,2
6	625,0	1,7	127,2
7	179,3	1,7	155,0
8	1940,5	2,4	227,9
9	3886,9	3,5	129,0
10	9170,3	6,9	541,1
11	3389,5	7,8	600,2
	1934,2	2,9	197,5

Статистическая модель, полученная по данной совокупности хозяйств, имеет следующий вид (множественная линейная регрессия):

$$Y = 67,1 + 3,2x_1 + 0,1x_2$$

Таким образом, коэффициенты эластичности: по основным фондам – 0,9, по трудовым ресурсам – 0,3.

Тогда производственная функция зависимости объемов производства от фондо- и трудообеспеченности будет иметь вид:

$$X^t = A\Phi^{0.9}L^{0.3}$$

где A - темп НТП; K - стоимость основных фондов; L - число занятых; α_1 - эластичность выпуска продукции по основным фондам; α_2 - эластичность выпуска продукции по труду.

Стоимость валовой продукции сельского хозяйства может быть «разложена» на составляющие (они в модели представлены в виде выходящих темпов): *Амортизация, Производственные затраты, Прибыль, Налог на продукт* (так как все налоги различаются по объекту, базе налогообложения и процентной ставке, здесь взята некая усредненная ставка – *Средняя ставка налога* – 20 % от стоимости валовой продукции), *Плата за труд*.

Амортизация служит источником пополнения основных фондов, которые, в свою очередь, определяют стоимость валовой продукции. Кроме этого источника основные фонды (см. одноименный уровень на схеме модели) пополняются за счет таких источников (ниже приводятся названия переменных модели): *Инвестиции из прибыли, Инвестиции из федерального бюджета, Инвестиции из бюджета* (здесь имеется ввиду региональный бюджет), *Заемные средства*. Исходя из стоимости основных средств предприятие платит налог на имущество (см. исходящие темпы из уровня *Основные фонды*). Другим исходящим темпом является *Потребление основных фондов*. Рассмотрим каждый из источников формирования основных фондов более подробно.

Прибыль формируется как разница между стоимостью валовой продукции и следующих составляющих – амортизация, материальные затраты (без амортизации), плата за труд, косвенные налоги. Часть прибыли идет на накопление.

Выражения модели приведены в **Приложении 5**.

Описанные выше закономерности реализованы с помощью инструментария визуального моделирования в системе Vensim, схема модели приведена на Рисунок 118.

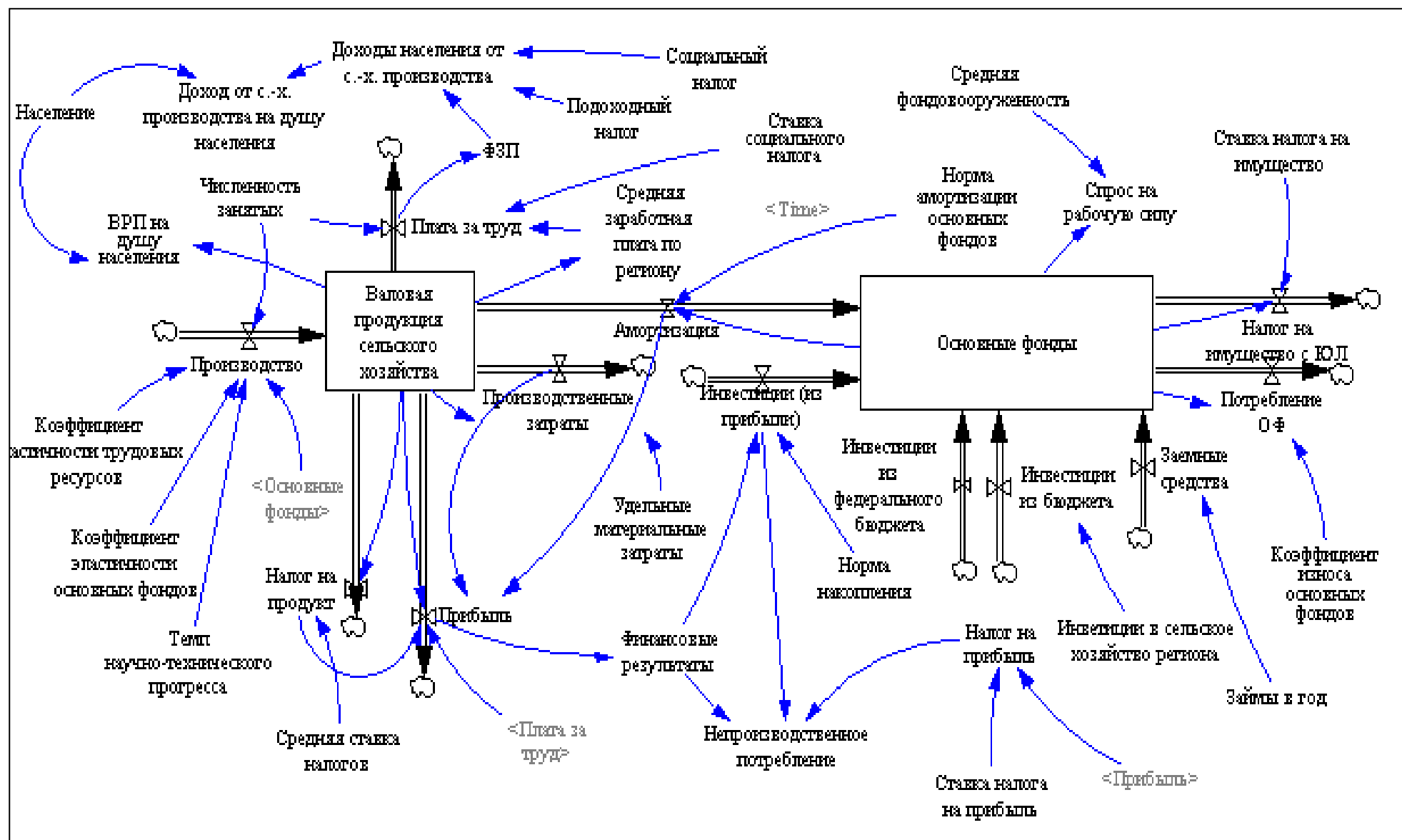


Рисунок 118 – Схема модели «Прогнозирование объемов регионального валового производства»

Результаты прогона модели представлены в **приложении 6**. Представленные на Рисунок 100 отдельные результаты прогона модели отражают то, что, например, с ростом эластичности по основным фондам (K_0) такие показатели, как *Объем валовой продукции сельского хозяйства* и *Доход от сельскохозяйственного производства на душу населения*, падают. Рост эластичности означает реакцию результативного показателя (в данном случае этих двух показателей) на изменение фондообеспеченности.

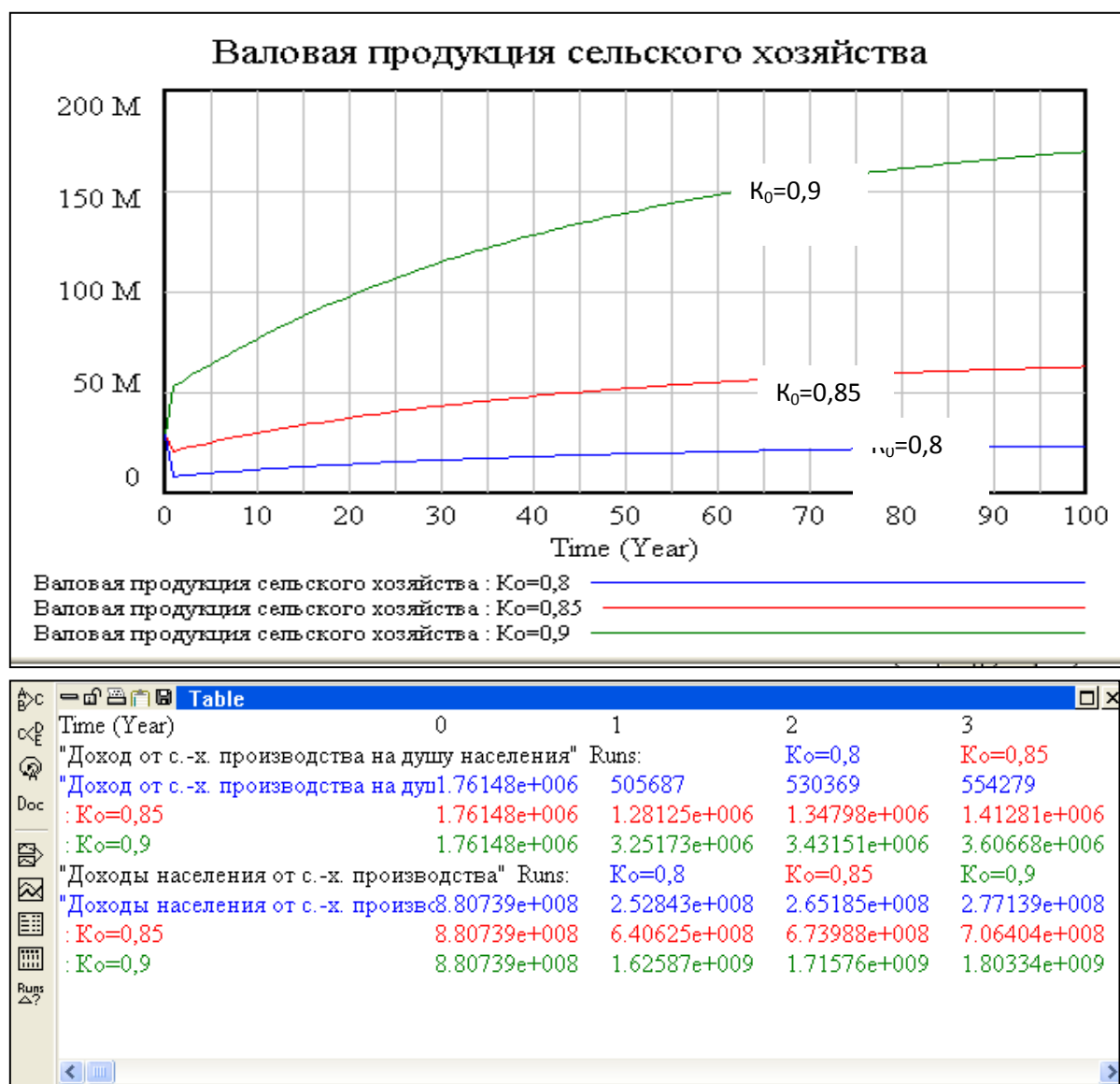


Рисунок 119 – Динамика валовой продукции сельского хозяйства административного района по результатам моделирования

Очевидно, что чем ниже фондообеспеченность в хозяйствах, тем более чувствительно производство к введению новой техники. Аналогичная картина складывается при увеличении эластичности производства по труду. Проведение се-

рии прогонов модели с такими изменяющимися характеристиками как коэффициенты эластичности по труду и капиталу, размеры инвестиций из федерального и регионального бюджетов показало следующее (табл.8).

Таблица 8 – Валовое производство в сельскохозяйственных организациях района, млн р. (по результатам моделирования: третий год с начала моделирования)

Инвестиции из федеральн. и ре- гион. бюджета, тыс. р. Эластичность ВП по фондам	260	270	280	290	300	310	320	330	340	350
0,5	38,4	41,2	43,8	45,2	47,8	50,2	52,0	53,6	54,5	55,3
0,55	38,9	41,8	44,4	45,8	48,5	50,9	52,7	54,4	55,3	56,1
0,6	39,5	42,4	45,0	46,5	49,1	51,6	53,5	55,1	56,0	56,9
0,65	40,0	43,0	45,7	47,1	49,8	52,3	54,2	55,9	56,8	57,7
0,7	40,6	43,6	46,3	47,8	50,5	53,1	55,0	56,7	57,6	58,5
0,75	41,2	44,2	47,0	48,5	51,2	53,8	55,7	57,5	58,4	59,3
0,8	41,7	44,8	47,6	49,1	52,0	54,6	56,5	58,3	59,2	60,1
0,85	42,3	45,4	48,3	49,8	52,7	55,3	57,3	59,1	60,1	61,0
0,9	42,9	46,0	49,0	50,5	53,4	56,1	58,1	59,9	60,9	61,8

Данные таблицы показывают, что с ростом обоих показателей – эластичность валового производства по фондам и объем инвестиций объем валовой продукции возрастает.

Контрольные вопросы к главе III

1. Сущность метода системной динамики.
2. Что представляет собой событийно-ориентированный подход к моделированию систем.
3. Что такое «обратные связи» в системе? Положительные и отрицательные петли обратной связи?
4. Привести примеры обратных связей в биологических и социально-экономических системах.
5. Что такое структура системы с обратной связью? Привести примеры усиления и запаздываний в системах с обратной связью.
6. Особенности моделирования сложных динамических систем
7. Этапы имитационного моделирования.
8. Что такое потоковые сети?
9. Объясните сущность основных элементов модели системной динамики – темпов, уровней, констант, вспомогательных переменных.
10. Виды уравнений в моделях системной динамики. Экзогенные переменные
11. Основное меню программы Vensim.
11. Как создать новую модель, установить горизонт моделирования?
12. Проведение прогона модели. Виды прогонов.
13. Виды окон в Vensim.
14. Функции панели управления моделированием (Control Panel).
15. Создание таблиц пользователя.
16. Состав панели инструментов для построения модели.
17. Пример построения модели в системе Vensim PLE. Редактор выражений.
18. Создание уровней, темпов, вспомогательных переменных модели, внутренних переменных.
19. Сущность и порядок проведения вычислительного эксперимента.
20. Функция LOOKUP: необходимость и порядок применения.
21. Оценка чувствительности имитационной модели.
22. Оптимизационный эксперимент с моделью.

ГЛАВА 4. АГЕНТНОЕ МОДЕЛИРОВАНИЕ

4.1. Сущность агентного моделирования процессов и систем

Агентное моделирование – это сравнительно новый метод имитационного моделирования. Если метод системной динамики существует около 70 лет, то агентное моделирование насчитывает немногим более 20 лет. Агентное моделирование считается наиболее мощным и универсальным инструментом, так как, так как оно позволяет учесть любые сложные структуры и поведения. Также, этот подход имеет преимущество, состоящее в том, что разработка модели возможна в отсутствии знания о глобальных зависимостях и влиянии элементов модели друг на друга. Однако, моделируя индивидуальную логику поведения участников процесса, при прогоне модели на выходе получают глобальное поведение системы.

Агентное моделирование можно определить как метод имитационного моделирования, исследующий поведение децентрализованных агентов и то, как это поведение определяет поведение всей системы в целом и, если это – система экономического характера – результативность ее работы. Целью агентного моделирования является то же, что и разработки других видов имитационных моделей – прогнозирование возможного развития системы.

В отличие от дискретно-событийного моделирования, где динамические объекты являются безликими, рассматриваемыми в виде аморфной массы, агентное моделирование имеет дело с агентами – объектами, имеющими индивидуальные свойства и сложные отношения между собой, не сводимые к простой обработке заявок, как в дискретно-событийном моделировании. То есть – это децентрализованные элементы. Агентное моделирование предполагает сосредоточение на индивидуальном поведении объектов и их взаимодействии. Оно наибольшим образом подходит для моделирования сложных бизнес-процессов, сложных взаимоотношений покупателей и продавцов на рынке, сложных социальных процессов и т.д. Агентное моделирование позволяет моделировать разрозненные элементы системы. Здесь моделируется поведение отдельных агентов, из взаимодействия которых рождается обобщенное поведение системы. То есть результатом взаимодействия агентов становится поведение системы в целом. Поэтому агентное моделирование называют моделированием «снизу-вверх». Одной из задач агентного моделирования является необходимость понять природу какого-либо сложного процесса или социального явления.

Агент – это некий динамический объект, который обладает автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением и другими агентами, а также может изменяться. Агент обладает автономностью и относительной независимостью: он со-

вершает определенные действия, вступает в контакт с другими агентами, принимает какие-то решения на основе анализа ситуации, перемещается в определенной среде и адаптируется в ней. В качестве агентов можно рассматривать работников предприятия, сами предприятия, поставщиков продукции, покупателей продукции на рынке (домохозяйства), непосредственно саму продукцию и др. Агентами могут быть не только отдельные индивидуумы, но и группы индивидуумов, выступающие как единые индивидуумы в принятии каких-либо решений. В реальности эти объекты имеют свои истории, цели, желания, качества и свойства. Так, работники различаются по уровням квалификации, возрасту, способностям и др. Предприятия, даже одной отрасли, различаются по множеству показателей и др.

Агент в модели должен обладать следующими характеристиками – набор определенных характеристик и правил, определяющих поведение агента и правила принятия решений.

- возможность независимых действий и принятия решений по взаимодействию с другими агентами.

- наличие определенной среды для действий агента, позволяющей ему взаимодействовать с другими агентами.

- наличие определенной цели, определяющей его поведение

- способность агента к самообучению на основе собственного опыта.

Таким образом агент обладает определенными характеристиками, правилами поведения, памятью, особенностями принятия решений и алгоритмом изменения правил поведения.

Для моделирования агентов в определенной среде моделирования агентов задают определенным образом, определяют их свойства, структуру, поведение и семантику, объединяют в некую сеть и определяют ее топологию.

При задании (создании) агентов необходимо задать следующие характеристики:

- память агента, общие характеристики, механизмы взаимодействия с другими объектами системы, связи с интерфейсами для сбора статистики о поведении объекта и других характеристиках модели;

- способ взаимодействия агента с окружающей средой;

- способ взаимодействия агентов друг с другом;

- способ задания поведения агента;

- способ изменения поведения агента (в ходе коммуникации с другими агентами или в результате пополнения знаний);

- механизм принятия решений согласно поставленной цели и задачам агента;

- механизм действий агента в условиях неопределённости;

- механизм пополнения знаний агента во время прогона модели;
- средства адаптации к среде моделирования, под влиянием которой агент изменяет свою внутреннюю структуру, поведение и семантику;
- механизм продвижения модельного времени;
- механизмы сбора статистики о состоянии агента и состоянии системы.

Различия агентных моделей и моделей системной динамики

Агентные модели определенным образом отличаются также и от моделей системной динамики. В таблице 9 дана краткая сравнительная характеристика этих концепций моделирования.

Таблица 9 – Сравнительная характеристика метода моделей системной динамики и агентных моделей

Характеристики	Модели системной динамики	Агентные модели
Основной элемент модели	Петля обратной связи	Агент
Взаимосвязь элементов модели	Потоки различной природы (материальные, человеческие, денежные, информационные) взаимодействуют между собой	Агенты взаимодействуют друг с другом и другими автономными объектами модели
Предмет моделирования	Структура системы	Правила поведения агента
Направление моделирования	«Сверху-вниз»	«Снизу-вверх»
Способ продвижения модельного времени	В соответствии с выбранным шагом модельного времени, с непрерывным (плавным изменением величин элементов модели)	В соответствии с выбранным шагом модельного времени, дискретно
Аппарат, лежащий в основе моделирования	Дифференциальные и интегральные уравнения	Логика (поведение) элементов модели

Если в моделях системной динамики результат моделирования зависит от набора элементов модели, структуры системы (взаимодействия потоков и накопителей) и заданного характера динамики элементов модели, то в агентном моделировании поведение системы формируется за счет закладываемых правил поведения агентов.

Как было сказано выше, правила поведения и взаимодействия агентов могут меняться в результате самообучения агента и изменения характера принимаемых им решений. В отличие от моделей системной динамики, агентные модели имеют возможность адаптации – изменению структуры модели с целью более эффективного функционирования в окружающей среде. Вместе с тем, в моделях системной динамики при их неизменной структуре действие петель обратных связей определенным образом балансирует величины элементов модели. И в этом смысле можно также говорить о возможности некоторой адаптации (уравновешивании, стабилизации) модели в виде изменяющихся значений ее элементов.

Если в системной динамике источником нелинейности динамики элемента системы является аккумуляция потоков, вызванная эффектами запаздывания, то в агентном моделировании так называемые «события» могут вызвать изменения в поведении системы. Например, в модели работы ремонтной бригады, обслуживающей энергетические установки, выход из строя одной из них меняет бизнес-процесс в плане маршрута передвижения бригады, времени задержки на обслуживание и других характеристик. Фигурально модели системной динамики можно сравнить с моделированием роста леса, а агентные модели – с моделированием роста деревьев.

Поскольку причинно-следственные связи, особенно в социальных или бизнес-процессах, бывают довольно сложными, их сложно отразить с помощью агентного моделирования. В таких случаях применяют смешанный подход. Так, например, при моделировании динамики потребительского рынка с помощью элементов системной динамики моделируются взаимосвязи (например, цена-спрос, доходы населения – спрос и др.), а с помощью элементов агентного моделирования – индивидуальное поведение потребителей, их предпочтения. То же самое относится и к соединению в одной модели агентного моделирования и дискретно-событийного (пример такой модели будет рассмотрен ниже).

Структура системы агентного моделирования

В общем виде систему агентного моделирования можно представить следующим образом (Рисунок 120).

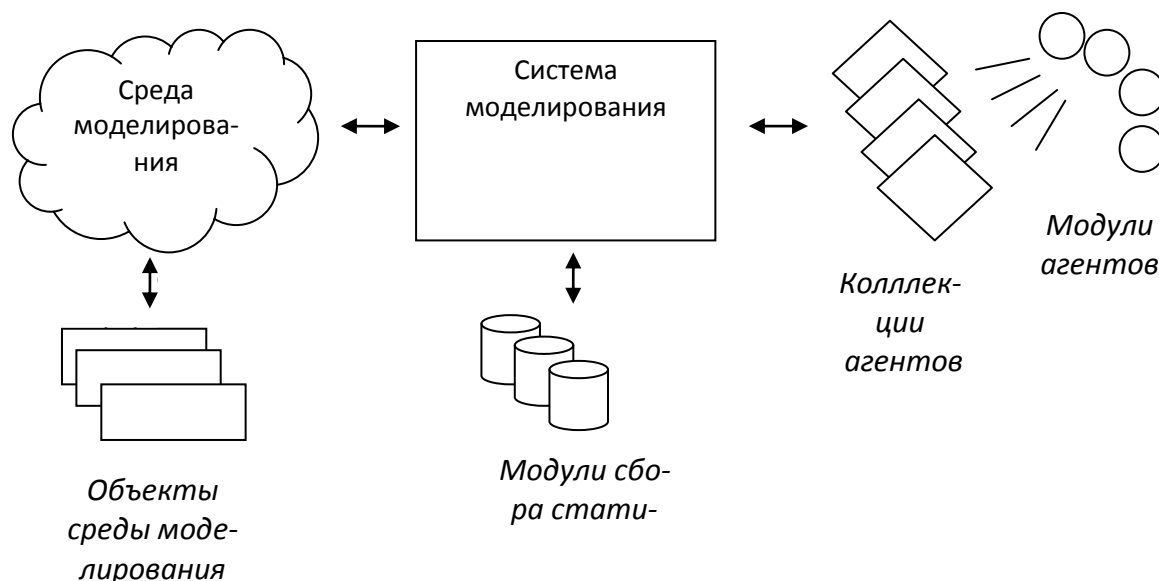


Рисунок 120 – Структура системы агентного моделирования

Агенты и модули агентов. Агенты представляются в виде объектов, которые используют интерфейсы, предлагаемые ядром и системными компонентами. Агенты также представляют собственные интерфейсы для контроля и сбора статистики.

Архитектура агента включает центральный компонент-контроллер, занимающийся процессом жизнедеятельности агента. Получив управление, он обращается к рецепторам с целью получения информации о среде моделирования. Затем контроллер, основываясь на полученных данных и данных из памяти агента, передаёт присоединяемым внешним компонентам, отвечающим за структуру, поведение и семантику агента (базы знаний, механизмы принятия решений, механизмы логического вывода, модули сбора информации для настройки весов нейросетей, анализаторы данных, модули управления агентом в условиях неопределённости и т.д.). Обычно они представляются в виде функциональных блоков, т.е. содержат вход, выход, подсистему управления и средства реализации. После отработки всех модулей агент с помощью эффекторов предпринимает активные действия (например, движение в некотором направлении). Информация обо всех действиях передаётся системе управления через диспетчер. Таким образом, диспетчер служит связующим звеном агента с системой моделирования. Согласно принципам открытых систем, модули должны иметь доступ к данным и процедурам, доступным на уровне агента (информация о среде моделирования, онтология предметного уровня, функции системы управления). Также они сами должны предоставить доступ к собственным интерфейсам (в частности, для сбора статистики). По сути, модули сбора статистики представляют собой «системных аген-

тов», которым доступна вся информация обо всех агентах и среде моделирования. Помимо собственно агентов, они также имеют доступ к зарегистрированным внешним модулям агентов, которые упоминались выше. Модули сбора статистики не должны жёстко привязываться к системе управления. По возможности, следует реализовать механизм регистрации, чтобы обеспечить возможность добавления новых сборщиков. Онтологии Поскольку в разрабатываемой системе подразумевается инвариантность моделирования агентов от предметной области, то следует каким-либо образом определить эту предметную область, чтобы пользователь, агенты, система управления, модули сбора статистики и прочие компоненты оперировали общими понятиями. В этом случае рекомендуется применять онтологический подход, т.к. он является общепризнанным средством спецификации предметной области.

Агентные модели включают следующие компоненты:

- отдельные агенты;
- правила принятия решений и правила для обучения агентов (их адаптация);
- правила взаимодействия между агентами;
- внешняя среда.

Взаимодействие между объектами классов осуществляется посредством входных и выходных переменных (непрерывно), а также через механизм портов (дискретно) (Рисунок 121).



Рисунок 121 – Взаимодействие между объектами классов

При разработке модели посредством концепции агентного моделирования разрабатываются классы активных объектов, которые могут содержать в себе другие объекты. Отдельные экземпляры объектов при запуске модели представляют собой дерево, вершиной которого является объект «root». Объекты имеют простые, глобальные и динамические параметры. Также иногда создаются сово-

купности объектов одного класса (replication). При разработке модели имеется возможность создавать и удалять объекты, а также связи между ними.

Вначале создается класс активного объекта, который задает внутреннюю структуру агента, а затем создается необходимое количество экземпляров этого класса, каждый из которых представляет отдельный агент модели.

Затем в модель добавляют агентов путем добавления объекта класса, задающего структуру агента (Person), в класс, моделирующий окружение агентов (Main). Характеристики агента задаются с помощью *параметров* класса. Все агенты обладают общей структурой, поскольку все они задаются объектами одного класса. Параметры позволяют задавать характеристики индивидуально для каждого агента.

Затем визуально создается поведение агента с помощью *стейтчарта* (диаграммы состояний). Стейтчарт – это начальное состояние, о чем свидетельствует указатель начального состояния, направленный в это состояние. Если стейтчарт будет находиться в этом состоянии, то это означает, что запланированное действие, которое должен совершать агент, пока еще не произошло. Затем добавляется еще одно состояние, которое означает, что агент уже совершил данное действие. Эти два состояния соединяются переходом, моделирующим действие.

Модель выполняется в соответствии с набором установок, называемым *экспериментом*. Можно создать несколько экспериментов и изменять рабочую конфигурацию модели, меняя текущий эксперимент модели (в дереве рабочего проекта эксперименты отображаются под элементом *Эксперименты*. Один эксперимент (*Simulation*), создается автоматически. Это *простой эксперимент*, позволяющий отображать анимацию и запускать модель с откалиброванными значениями параметров. Существуют и другие типы экспериментов (оптимизационный эксперимент, эксперимент для варьирования параметров), используемые в тех случаях, когда параметры модели играют существенную роль, и требуется проанализировать, как они влияют на поведение модели, или когда нужно найти оптимальные значения параметров модели.

Области применения агентного моделирования

Благодаря перечисленным выше особенностям и возможностям агентного моделирования, оно может использоваться и используется для моделирования самых различных областей – как экономики, так и социальной жизни общества.

- Решение проблем управления производством, маркетингом, конкуренцией – для моделирования производственных и логистических процессов на предприятии и оптимизации управления, поведения предприятия на рынке, в том числе – в конкурентной среде, сценарного планирования и др.

- Динамика и объемы товарных рынков, процессов страхования и налогообложения и др.

- Развитие общества в целом (исторические процессы).
- Компьютерные сети.
- Социальные (некомпьютерные) сети (например, террористические организации).
- Трафик пешеходного и транспортного движения.
- Поведение человека и здравоохранение (поведение человека в толпе и чрезвычайных ситуациях, взаимодействие различных систем организма человека, распространение эпидемий и др.).
- Биология и экология (поведение животных, развитие агробиоценозов и др.).

Наиболее часто агентные модели используются для моделирования поведения потребителей. При этом используются данные опросов клиентов, результаты маркетинговых исследований, сегментация рынка, таймеры очередей, частота прихода покупателей в магазин и данные о производительности, связанные покупки и др.

Еще одним из перспективных направлений применения агентного моделирования является управление коллективом организации в плане управления конфликтами, когда изучаются процессы возникновения конфликтов и выявляются наиболее экономически- и социально-эффективные методы разрешения конфликта.

Агентное моделирование целесообразно применять не только для управления рисками, так как рискованные события воздействуют на деятельность людей, на принимаемые управленческие решения.

Основными этапами агентного моделирования являются:

- постановка задачи (формулировка проблемы);
- выбор среды моделирования, позволяющей реализовывать поставленную задачу;
- разработка модели (задание шага модельного времени, определение агентов и задание их свойств (характеристик – атрибутов и правил поведения, задание среды их функционирования);
- прогоны и верификация модели (проверка ее адекватности);
- проведение экспериментов с моделью;
- анализ результатов;
- разработка предложений по совершенствованию деятельности объекта управления.

4.2. Имитационная модель уборки урожая в сельскохозяйственном предприятии

4.2.1. Постановка задачи

Экономическая эффективность работы предприятия, во многом определяется организацией бизнес-процессов. Для сельскохозяйственного предприятия наиболее напряженными периодами являются

Эффективность деятельности предприятия во многом определяется эффективностью организации производственных процессов. В сельскохозяйственном предприятии наиболее трудоемок и сложен в плане организации, материально-денежных и трудовых затрат процесс уборки сельскохозяйственных культур. Разработки имитационной модели процессов и проведения экспериментов с ней позволит оптимизировать параметры данного процесса.

Процесс уборки зерновых культур предполагает следующую последовательность операций. Комбайны и автомобили из гаража выезжают на поле, где комбайны производят уборку и обмолот зерновых. При заполнении бункера комбайн останавливается и выгружает зерно в подъезжающий автомобиль. После этого комбайн продолжает движение и уборку. При заполнении автомобиль направляется на склад, где происходит разгрузка зерна. Далее автомобиль направляется на поле и ожидает следующей остановки комбайна. В пункте приема зерно взвешивается и разгружается, затем отправляется на очистку и сушку, после чего отправляется на хранение в зернохранилище.

Результаты решения задачи позволяют проанализировать качество бизнес-процесса взаимодействия комбайнов, машин и пункта приема зерна, а также узнать, сколько времени уходит на уборку урожая, позволяет узнать, сколько времени машины тратят на простой в очередях и на ожидание комбайна, и сколько необходимо времени обслуживающим устройствам на обслуживания транзакта, а также затраты на уборку зерна, оптимальное время уборки зерна и оптимальное количество техники, а также выявить узкие места в данном процессе и оптимизировать работу системы.

4.2.2. Порядок создания модели

Для того чтобы создать новую модель, на панели управление выбирается опция создания модели, которое открывает окно «Новая модель». Далее указывается имя модели («Уборка зерновых культур»), местоположение хранения файлов модели. В качестве единицы модельного времени устанавливаются минуты (Рисунок 122).

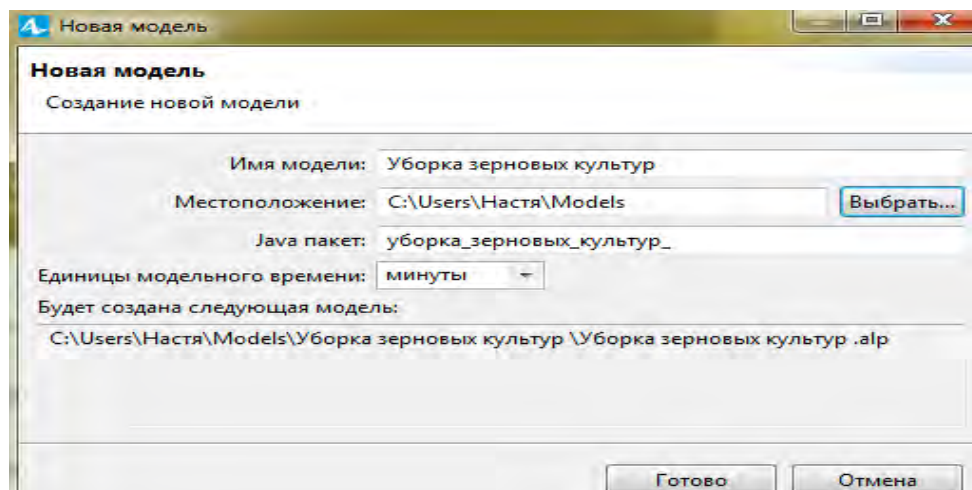


Рисунок 123 – Окно создания новой модели

При этом автоматически добавляется один тип агента Main, где будет отражаться работа модели и простой эксперимент *Simulation*, необходимый для прогона модели. Далее создается пространство, в котором будут функционировать все объекты модели (Рисунок 124).



Рисунок 124 – Создание пространства функционирования объектов модели

4.2.2.1. Создание агентов

Агент в моделировании – это элемент модели, который имеет индивидуальное поведение, параметры, переменные и т.д. Агентами могут представляться люди, товары, транспортные средства, здания, города, и т.д.

В данной модели будут присутствовать следующие агенты: два комбайна, два грузовых автомобиля и зерновой ток.

Для создания нового агента, на вкладке палитры выбирается *Агентное моделирование*, автоматически открывается окно создания агента, (Рисунок 125), где нужно выбрать соответствующую опцию для создания агента.

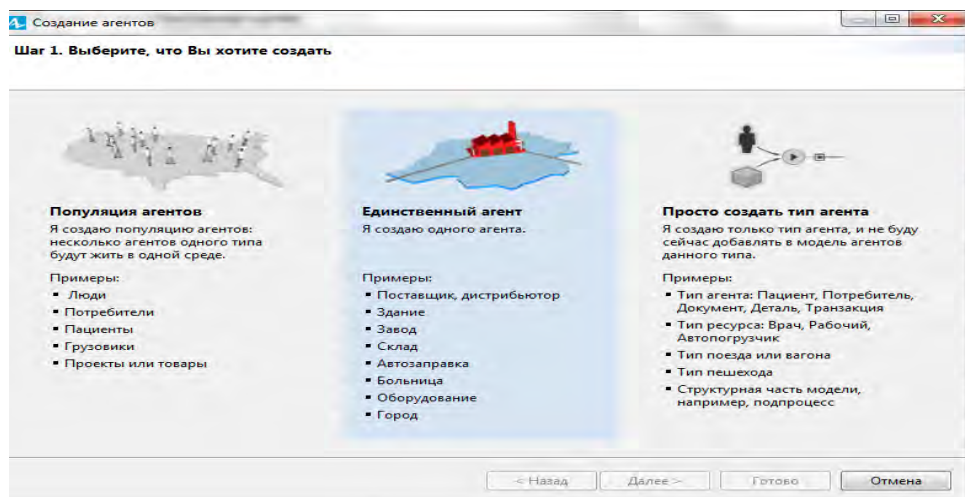


Рисунок 125 – Выбор опции создания агента

Следующим шагом создается тип агента комбайна. Так как создается новый тип агента, предлагаемые шаблоны не используются. Указывается имя типа – «Combine1», а имя агента – «combine1» (анимация комбайна здесь не задается, так как она будет создаваться позже вручную), процесс создания агента завершается.

По аналогии создаются все остальные агенты (Рисунок 126):

1. Combine1 – комбайн ПОЛЕСЬЕ;
2. Combine2 – комбайн ДОН1500;
3. Car1– грузовой автомобиль, в который зерно будет выгружаться из комбайна 1;
4. Car2 – грузовой автомобиль, в который зерно будет выгружаться из комбайна 2;
5. Bin (Stock) – зерновой ток.

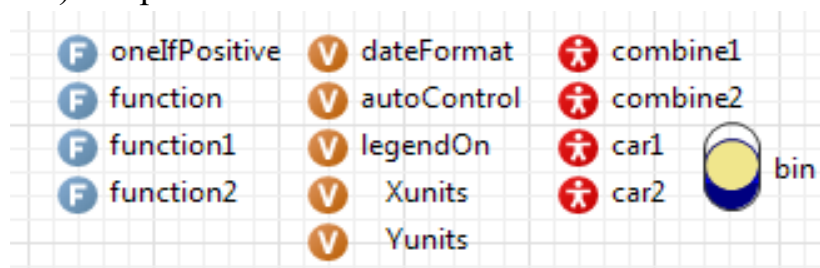


Рисунок 126 –Агенты модели

4.2.2.2. Настройка анимации агентов

Для настройки анимации в палитре выбирается вкладка *Презентация*. Размер объекта устанавливается с помощью линейки (Рисунки 127 и 128).

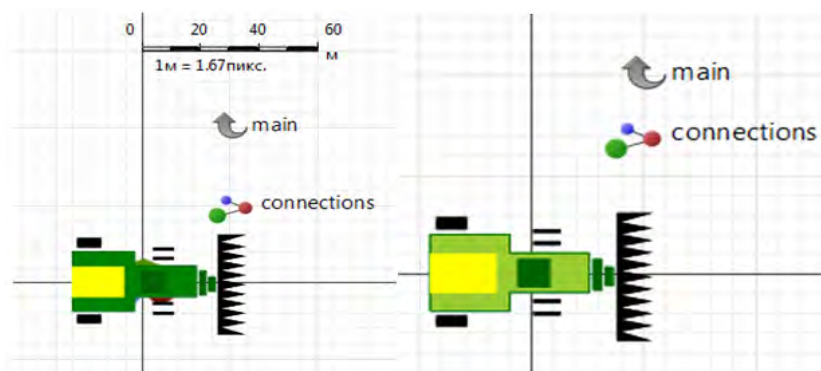


Рисунок 127 – Создание анимации агентов *combine1* и *combine2*.

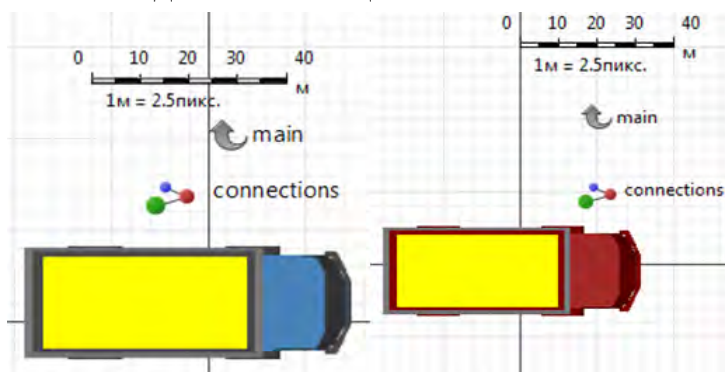


Рисунок 128 – Создание анимации агентов *car1* и *car2*

Агенты комбайна имеют ряд необходимых для его функционирования параметров, переменных и функций (Рисунок 129).

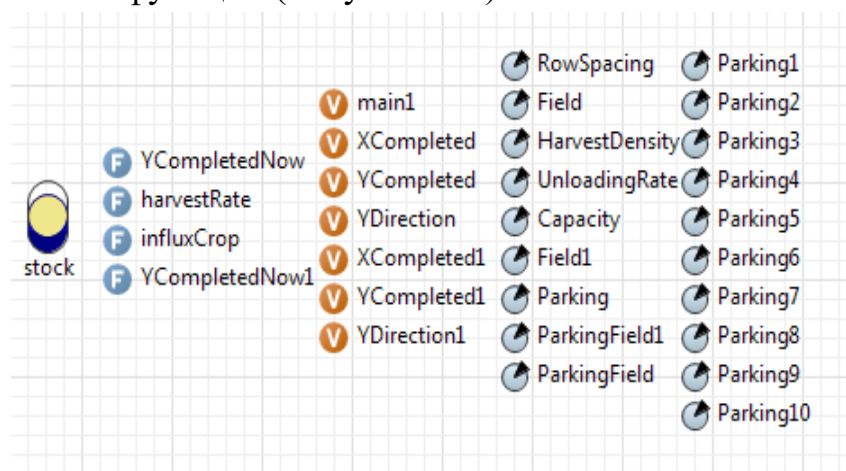


Рисунок 129 – Параметры агентов *combine1* и *combine2*

Данные параметры имеют следующие смысловые значения:

- 1) *RowSpacing* (Междурядье) – ширина захвата жатки комбайна;
- 2) *HarvestDensity* (Плотность урожая) – показывает среднюю плотность посевов зерновых культур;
- 3) *UnloadingRate* (Интенсивность разгрузки) – необходимое время на выгрузку зерна из бункера комбайна в машину;
- 4) *Capacity* (Объем) – объем бункера комбайна;
- 5) *Field* и *Field1* (Поле) – поля, убираемые данным комбайном;
- 6) *Parking* и *ParkingField1* – места прибытия агентов.

Агент содержит следующие переменные:

- 1) *main1* – переменная, которая обращается к главному агенту модели Main. Начальное значение: *get_Main()*;
- 2) *XCompleted, XCompleted1* – часть поля, уборка которого завершена (по оси X);
- 3) *YCompleted, YCompleted1* – часть поля, уборка которого завершена (по оси Y);
- 4) *YDirection, YDirection1* – направления движения комбайнов по полю. Начальное значение: NOTSTARTED.

Функции для работы агента:

- 1) *YCompletedNow, YCompletedNow1* – часть поля, на котором была завершена уборка.

Тело функции *YCompletedNow*:

```
if(inState( ПеремещениеПриСборе ))
    return getY() - Field.getY();
return YCompleted;
```

Тело функции *YCompletedNow1*:

```
if(inState( ПеремещениеПриСборе ))
    return getY() - Field1.getY();
return YCompleted1;
```

- 2) *harvestRate* – приток зерна в бункер комбайна. Тело функции: **return** *this.getSpeed(MPS) * scale.toLengthUnits(RowSpacing, METER) * HarvestDensity*.

На Рисунке 130 показаны функции и параметры, необходимые для отражения функционирования грузовых автомобилей.



Рисунок 130 – Параметры агента *car1* и *car2*

Агент *tractor* имеет следующие параметры:

- 1) *Speed* (Скорость) – показывает скорость движения агента***;

- 2) *Bin* – место остановки трактора в зернохранилище для разгрузки зерна;
- 3) *UnloadingRate* – интенсивность выгрузки зерна из машины в зернохранилищ (8 минут) е;
- 4) *Capacity* – объем кузова автомашины (6 м³);
- 5) *ParkingField* – *Parking7* – места прибытия агента в пространстве.

Агент имеет переменную *main1*, которая необходима для связи с главным агентом, а также функцию *moveWithCombine1*, которая необходима для взаимодействия трактора с комбайном. Тело функции:

```
//synchronize position
jumpTo( main1.combine1.getX()-40, main1.combine1.getY() );
//synchronize movement
if( main.combine1.isMoving() ) {
    double x = main1.combine1.getTargetX() - 40;
    double y = main1.combine1.getTargetY();
    moveTo( x, y );}
```

4.2.2.3. Настройка логических процессов

Если у агента можно выделить несколько состояний, выполняющих различные действия при происхождении каких-то событий, или если у агента есть несколько качественно различных поведений, последовательно сменяющих друг друга при происхождении определенных событий, то поведение такого объекта может быть описано в терминах диаграммы состояний. Диаграмма состояний позволяет графически задать пространство состояний алгоритма поведения объекта, а также события, которые являются причинами срабатывания переходов агента из одних состояний в другие, и действия, происходящие при смене состояний. Для того, чтобы создать диаграмму состояний на панели *Палитра* выбирается палитра *Диаграмма состояний*.

Вначале создадим диаграмму, которая будет описывать управление бункером комбайна (Рисунок 131).

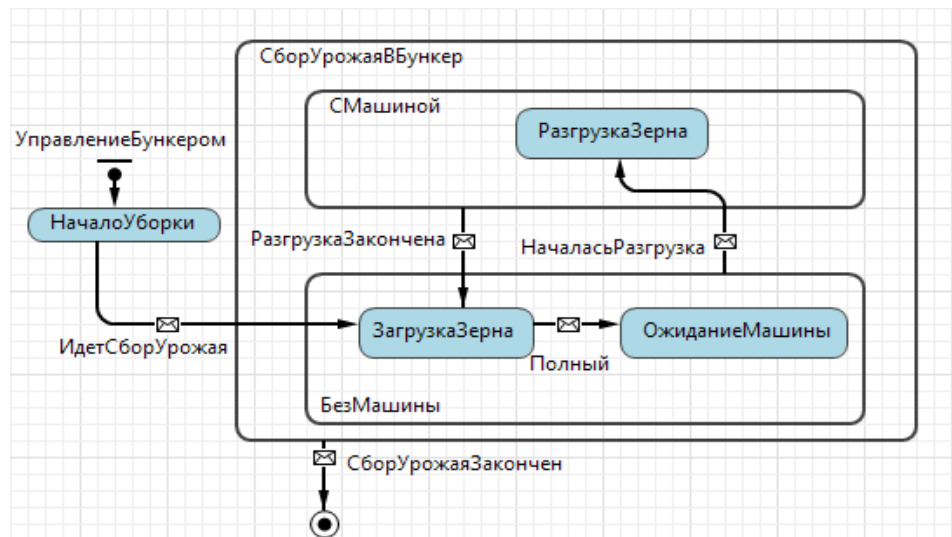


Рисунок 131 – Диаграмма управления бункером комбайна

Диаграмма показывает, что при обращении к началу диаграммы состояния (*УправлениеБункером*) начинается уборка зерна (состояние *НачалоУборки*), после чего при получении сообщения «*STARTED_HARVESTING*» агент переходит в состояние *ЗагрузкаЗерна*, то есть комбайн начинает загружать зерно в бункер. В данном состоянии прописываются действия агента при входе и выходе (Рисунок 132).

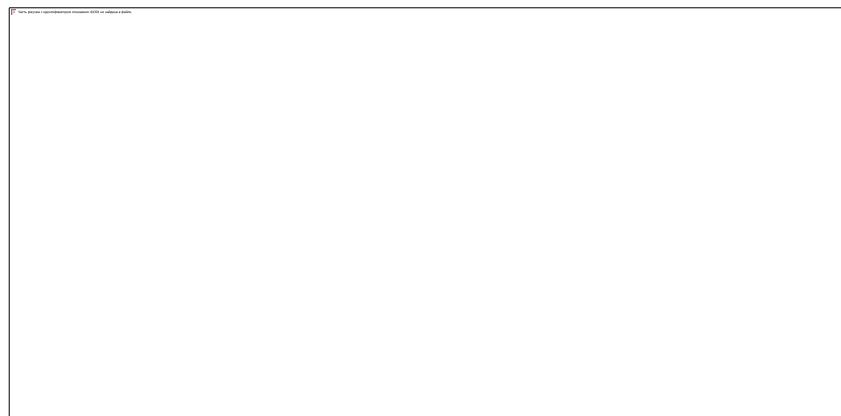


Рисунок 132– действия агента в состоянии «ЗагрузкаБункераЗерном»

Как только бункер будет заполнен, должно прийти сообщение «*FULL*», которое вызовет действие остановки комбайна, для этого необходимо прописать следующее действие (Рисунок 133).

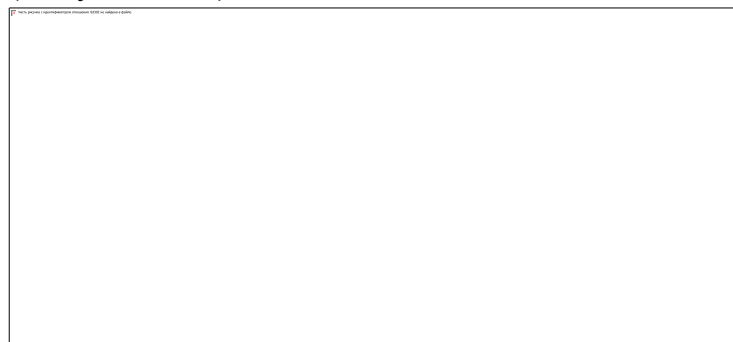


Рисунок 133 – Переход агента при полном бункере

Далее комбайн ожидает машину, как только машина достигнет комбайна приходит сообщение «STARTED_UNLOADING», после чего агент переходит в состояние «ВыгрузкаЗерна» и выполняет следующие действия (Рисунок 134).

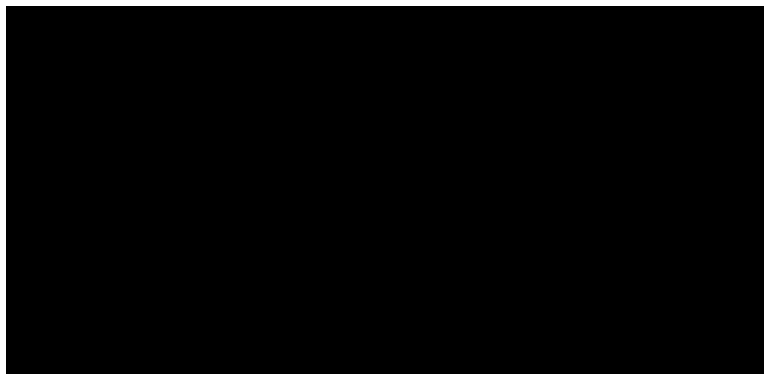


Рисунок 134 – Действия агента в состоянии «ВыгрузкаЗерна»

Как только выгрузка зерна будет закончена, агент снова переходит в состояние «ЗагрузкаБункераЗерном», при этом объявляя о начале движения комбайна (Рисунок 135).

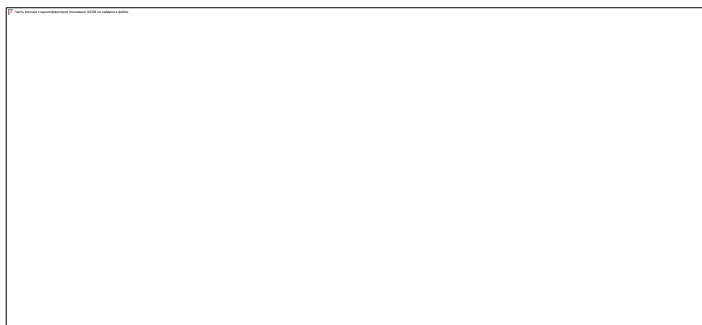


Рисунок 135 – Объявление агенту о начале движения.

После завершения работы с диаграммой состояния по управлению бункером, необходимо создать диаграмму, которая будет описывать движения комбайна в пространстве (Рисунок 136 и 137).

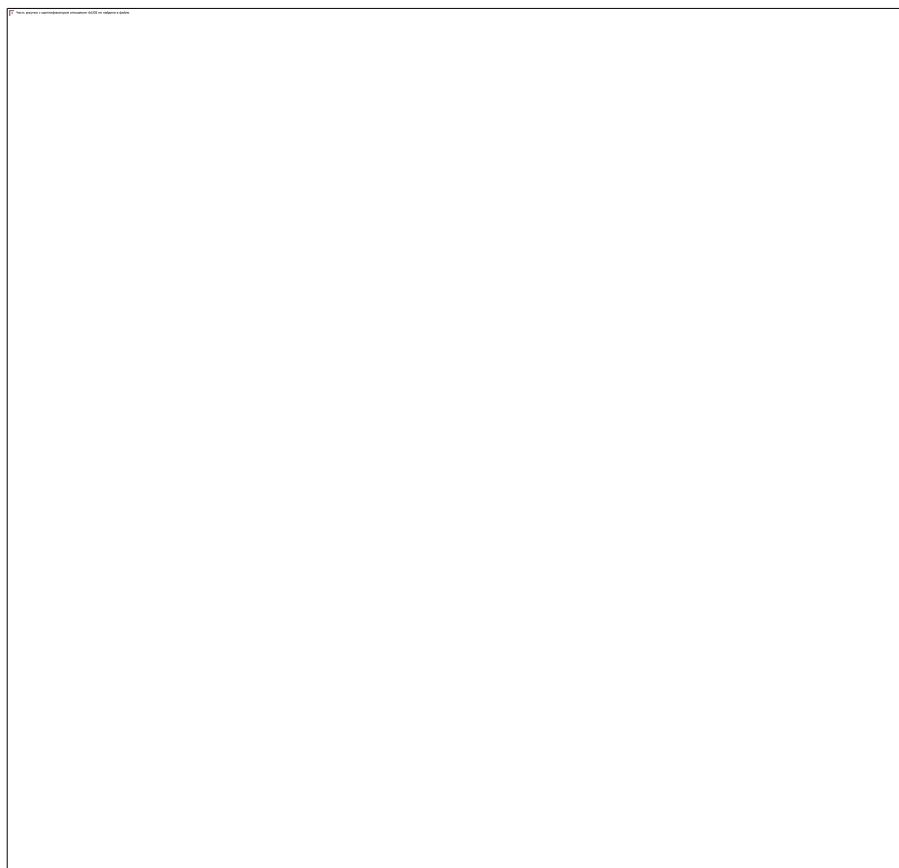


Рисунок 136 – Диаграмма движения агента *Combine1* в пространстве

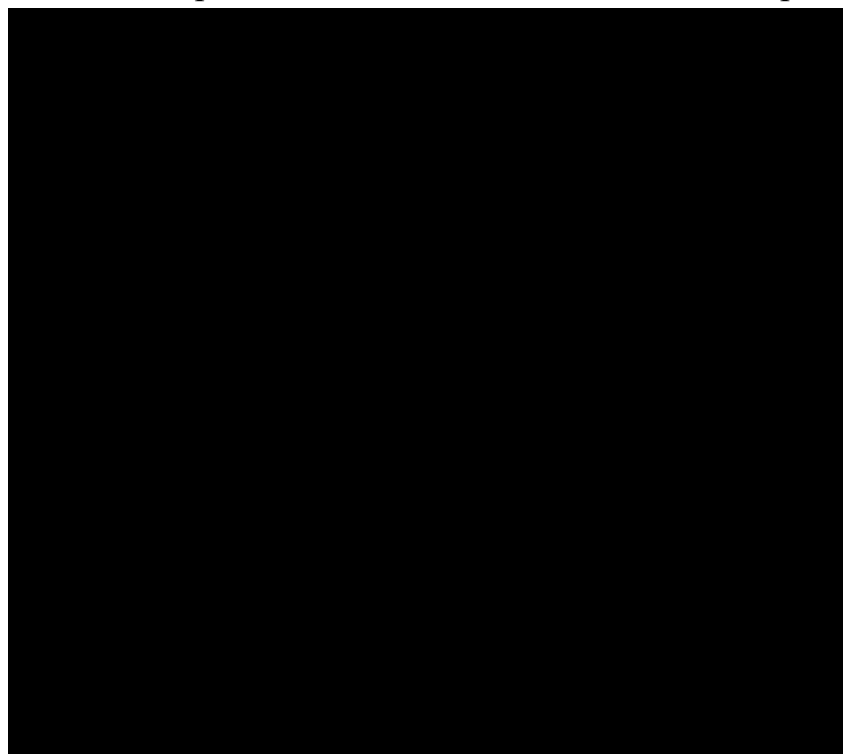


Рисунок 137 – Диаграмма движения агента *Combine2* в пространстве

В свойствах агента прописываются условия действия агента при запуске и при достижении точки назначения (Рисунок 138).

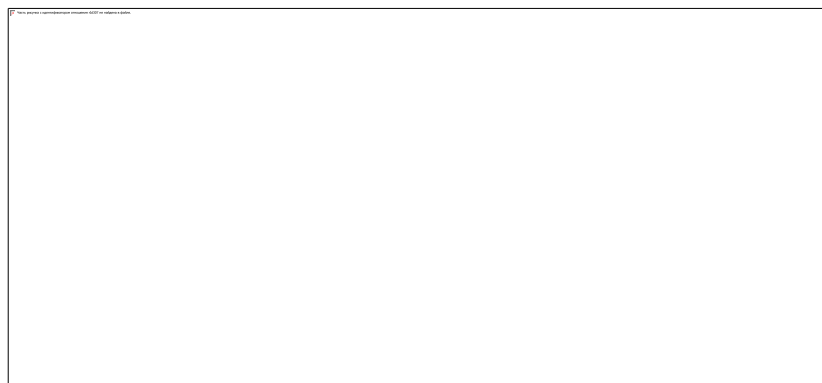


Рисунок 138 – Действия агента *Combine1*

При запуске модели агент переходит в состояние *ПрипаркованВГараже*, далее агент начинает движение к полю. Достигнув точки назначения, агент получает сообщение «*ARRIVED*» и переходит состояние *ПеремещениеПриСборе*, а также объявляет о начале загрузки зерна в бункер (Рисунок 139).

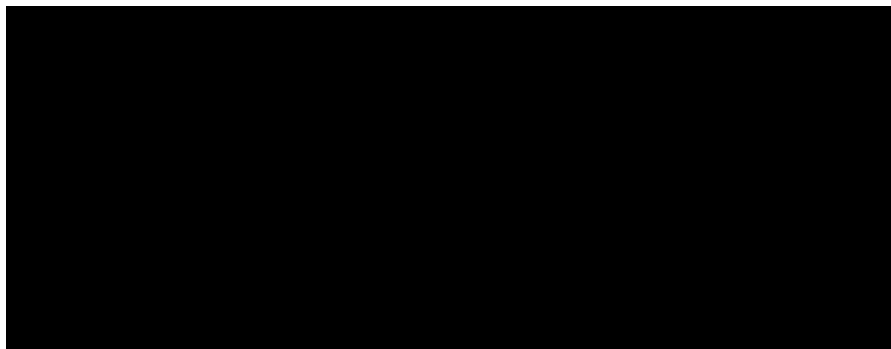


Рисунок 139 –Переход агента в состояние *ПеремещениеПриСборе*

Состояние *ПеремещениеПриСборе* включает в себя два состояния, которые определяют направление движения комбайна на поле. Как только бункер будет заполнен, агент получает сообщение «*STOP*» и переходит в состояние *ОжидаетМашину*, при этом объявляя о начале движения машины (Рисунок 140).



Рисунок 140 – Действие агента в состоянии *ОжидаетМашину*

Как только разгрузка будет завершена, приходит сообщение «*RESUME*» и комбайн продолжает движение.

Как только комбайн доходит до конца поля его необходимо направить на другое. Для этого создаем условие, с помощью которого агент перейдет в состояние *НаСледующееПоле* (Рисунок 141).

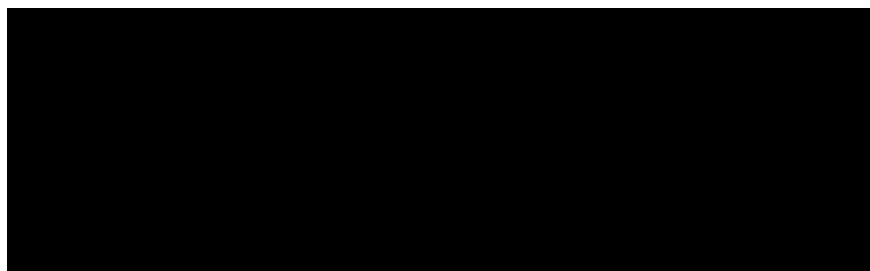


Рисунок 141 – Действие агента при завершении уборки первого поля

Движение комбайна при уборке второго поля задается же такое, как и при уборке первого. Как только второе поле будет убрано, ставится такое же условие, как и при завершении уборки первого поля, при выполнении которого комбайн направляется в гараж.

Затем диаграммы состояний управления бункером и движения первого комбайна копируются в агента второго комбайна.

Далее создается диаграмма состояний, которая будет описывать поведение трактора при уборке зерновых культур (Рисунок 142).

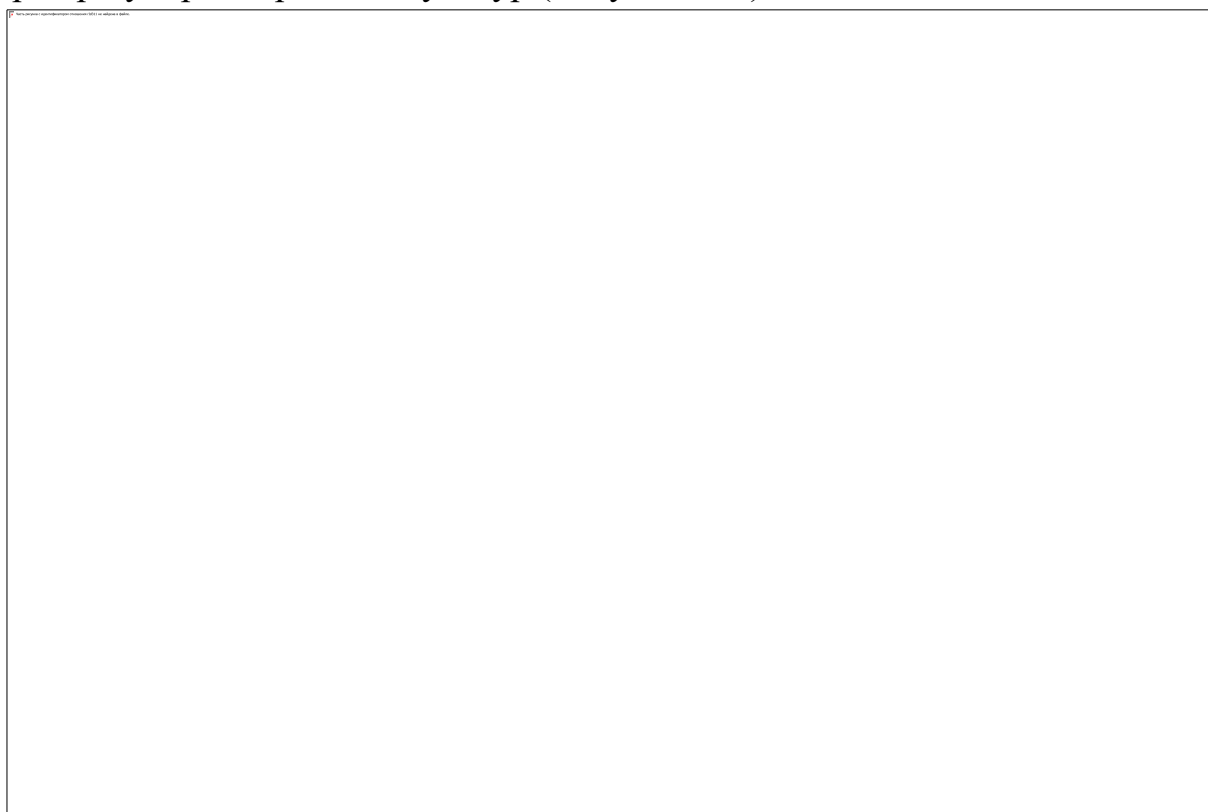


Рисунок 142 – Диаграмма движения агентов *Car1* и *Car2* в пространстве

Также как и комбайн, трактор при запуске модели находится в гараже, после чего начинает движение на поле. А поле достижения точки назначения переходит в состояние *ОжиданиеКоманды* (Рисунок 143).

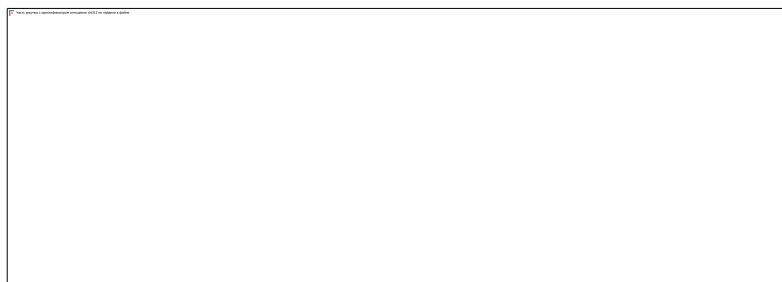


Рисунок 143 – Переход агента в состояние *ОжиданиеКоманды*

Далее после завершения загрузки зерна в бункер комбайна приходит сообщение «*GO_TO_COMBINE*», трактор переходит в состояние *ЕдетККобайну* (Рисунок 144).

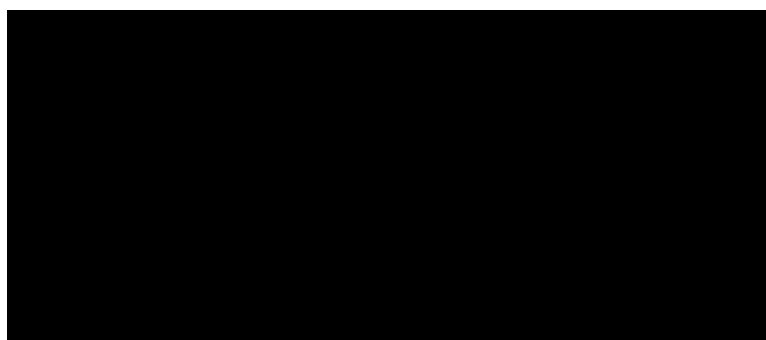


Рисунок 144 – Действие агента в состоянии *ЕдетККобайну*

Достигнув комбайна, трактор переходит в состояние *СКомбаном*, которое показывает их взаимодействие (Рисунок 145).

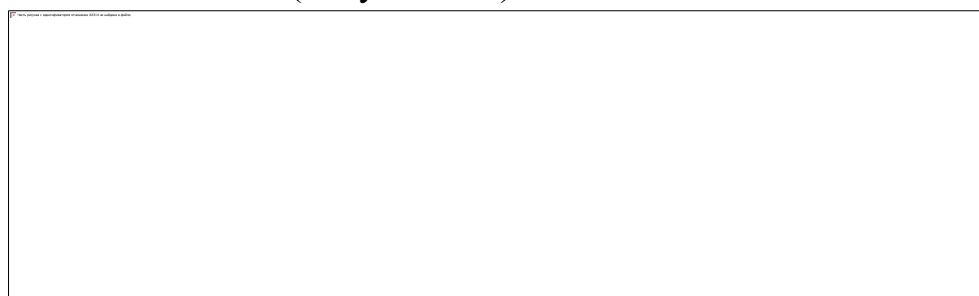


Рисунок 145 – Действие агента в состоянии *СКомбаном*

После завершения загрузки приходит сообщение «*GO_TO_BIN*» и трактор направляется в зернохранилище, достигнув точки назначения, переходит в состояние *РазгружаетЗерно* (Рисунок 146).

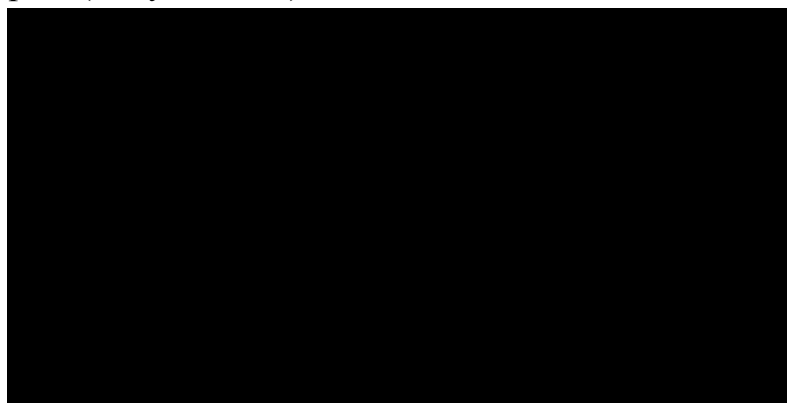


Рисунок 146 – Действие агента в состоянии *РазгружаетЗерно*

Агент *Stock* имеет следующие параметры:

- 1) *Capacity* – устанавливает объем бункеров агентов;
- 2) *InFlow* – показывает приток зерна в бункер;
- 3) *OutFlow* – показывает отток зерна из бункера;
- 4) *VInitial* – означает начало загрузки зерна в бункер;
- 5) *onFull* – является динамическим параметром и показывает состояние бункера, когда он полный;
- 6) *onEmpty* – также является динамическим параметром и показывает состояние бункера, когда он пустой (Рисунок 147).



Рисунок 147 – Параметры и свойства агента *Stock*

Также необходимо создать переменные *V*, *on*, *out*, которые необходимы для задания функций агента.

Событие *timer*, которое моделирует полноту бункера при разгрузке и загрузке зерна. Действие события:

```
if(InFlow>OutFlow) {  
    V = Capacity;  
    onFull();  
} else {  
    V = 0;  
    onEmpty();  
}
```

Также для правильного функционирования агента необходимо создать несколько функций:

- 1) *inow* – показывает начальное состояние бункера в зависимости от разгрузки или загрузки зерна.

Тело функции:

```
if (timer.isActive() ) {  
    if ( in > out )  
        return Capacity - timer.getRest() * ( in - out );  
    else
```



```

return timer.getRest() * ( out - in );
}
return V;
2)   setv – устанавливает количество зерна загружаемого в бункер.
Тело функции:
if( v < 0 || v > Capacity )
error ( "The value specified in setv() is outside [0,Capacity] range" );
timer.reset();
V = v;
if(InFlow>OutFlow ) {
timer.restart ( ( Capacity - V ) / ( InFlow - OutFlow ) );
} elseif (OutFlow>InFlow ) {
timer.restart( V / ( OutFlow - InFlow ) );
}
3)   Onchange – показывает изменение зерна в бункере.
Тело функции:
if(timer.isActive() ) {
    V = vnow();
timer.reset();
}
if(InFlow>OutFlow ) {
timer.restart( ( Capacity - V ) / ( InFlow - OutFlow ) );
} elseif(OutFlow>InFlow ) {
timer.restart( V / ( OutFlow - InFlow ) );
}
in = InFlow;
out = OutFlow;

```

4.2.2.4. Создание диаграмм процессов

На данном этапе необходимо задать динамику процесса уборки зерновых культур, создав диаграмму из блоков Библиотеки моделирования процессов из вкладки «Палитра» Каждый блок задает определенную операцию, которая будет производиться над проходящими по диаграмме процесса агентами.

Сначала создадим диаграммы, которые показывают взаимодействие машин и комбайнов, где транзактом будет комбайн, а обслуживающим устройством – машина (Рисунок 148).

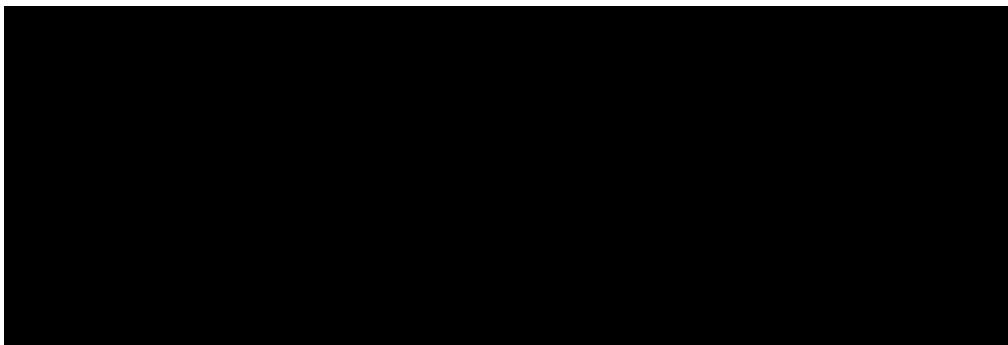




Рисунок 148 – диаграммы процессов выгрузки зерна в машины

На рисунке показано, что как только комбайн наполняет бункер зерном, он останавливается, это обозначено элементом  (Source), после остановки комбайн ожидает машину для перегрузки зерна, это ожидание элемент очереди



(Queue). Как только машина достигла комбайна, начинается выгрузка зерна, здесь машина является обслуживающим устройством – элемент  (Delay). После того как выгрузка закончилась, комбайн приступает к дальнейшей уборке, а машина направляется на зерновой ток.

Также необходимо создать диаграмму процессов выгрузки зерна на зерновой ток (Рисунок 149).

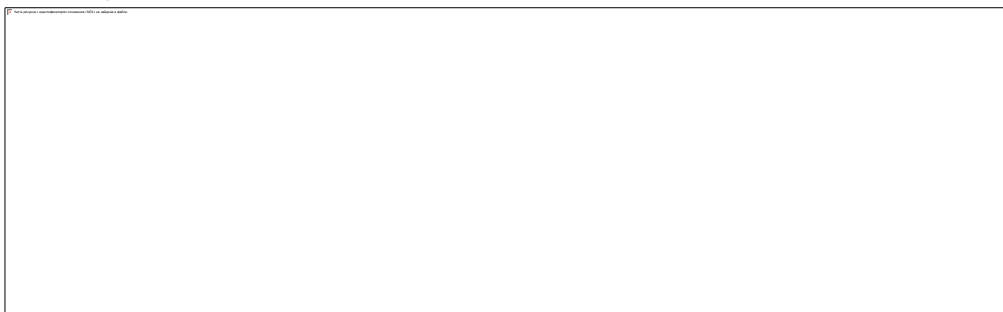


Рисунок 149 – диаграммы процессов разгрузки зерна на зерновой ток

На рисунке 3.28 видно, машины прибывают на зерновой ток, как только закончат загрузку зерна (Source Машина1 и Машина2). Далее машины ожидают разгрузки (QueueОчередь_на_разгрузку) и как только подходит время, начинается разгрузка зерна на зерновой ток (DelayЗерновой_ток). На данной схеме обслуживающим устройством является зерновой ток, а транзактом – партия зерна.

4.2.2.5. Добавление в модель элементов сбора статистики о характеристиках бизнес-процесса

Основными характеристиками бизнес-процесса уборки зерновых культур является время самой уборки зерна, а также время ожидания комбайнов машин и

время разгрузки зерна на склад. Также основными характеристиками являются коэффициент загрузки транспорта при уборке зерновых культур и зернового склада.

Эффективно организованный процесс уборки зерна осуществляется тогда, когда транспорт (комбайны и машины), а также пункт приема зерна загружены большую часть дня (коэффициент загрузки близок к 1) и когда время ожидания машин комбайнами и очередь на разгрузку зерна минимальны.

Для того чтобы выявить значения по данным характеристикам необходимо создать элементы статистики о характеристиках бизнес-процесса. Для этого необходимо выбрать палитру *Статистика* и добавить столбиковые диаграммы и временные графики (Рисунок 150).

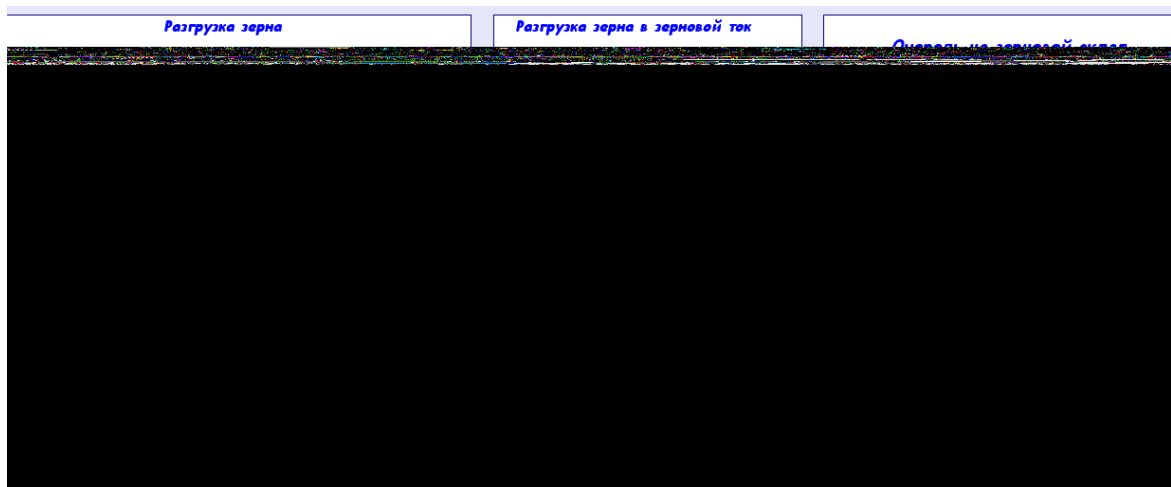


Рисунок 150 – Элементы статистики модели

В свойствах столбиковой диаграммы в поле «Заголовок» прописывается имя элементы, вводиться *Разгрузка_зерна1.statsUtilization.mean()* и *Разгрузка_зерна2.statsUtilization.mean()* в поле *Значение*. Здесь *Разгрузка_зерна1* и *Разгрузка_зерна2*- это имя объекта *Delay* [20]. У каждого объекта *Delay* есть встроенный набор данных *statsUtilization*, занимающийся сбором статистики использования этого объекта. Функция *mean()* возвращает среднее из всех измеренных этим набором данных значений (Рисунок 151).

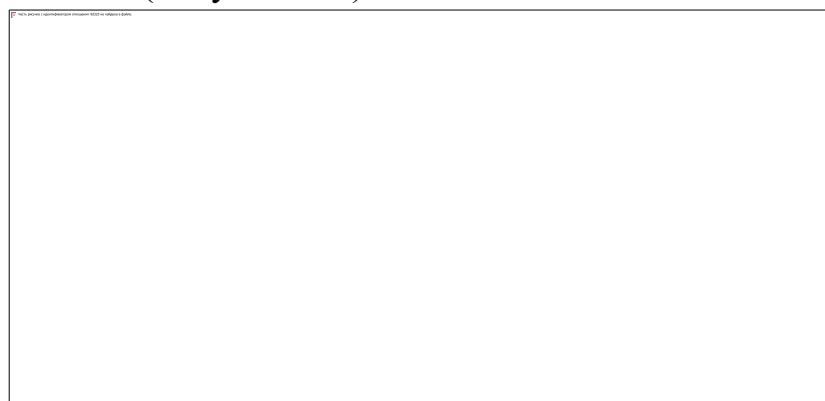


Рисунок 151 – Свойства столбиковой диаграммы

Во временных графиках прописывается значения *Ожидание_машины1.statsSize.mean()* и *Ожидание_машины2.statsSize.mean()*. Здесь *statsSize* - это имя объекта типа *СтатистикаStatisticsContinuous*, производящего сбор статистики очереди объекта *Queue* (Рисунок 152).

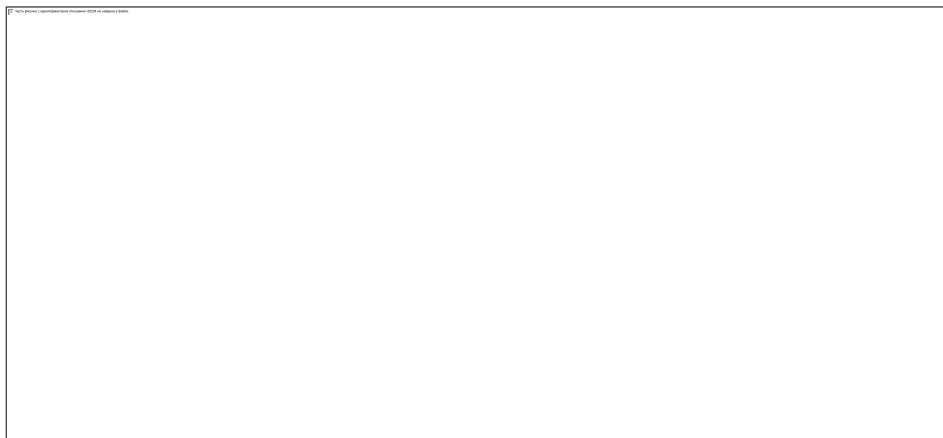


Рисунок 152 – Свойства временного графика

Далее по аналогии с временными графиками создается столбиковую диаграмму, которая будет показывать очередь на разгрузку зерна на склад (Рисунок 153).

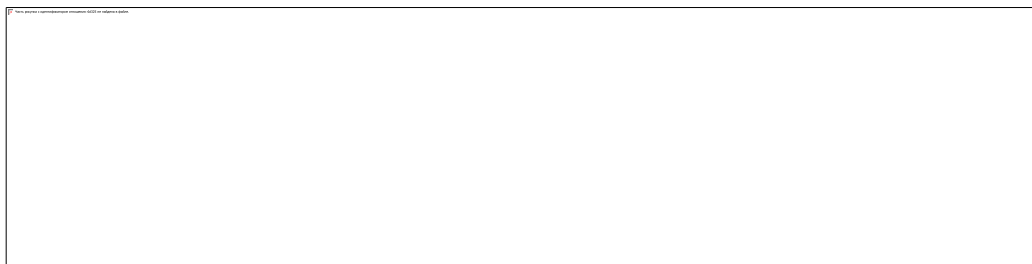


Рисунок 153 – Свойства диаграммы очереди на разгрузку

3.2.2.6. Создание элементов управления и прогон модели

Для удобства работы с моделью во время прогонов, необходимо создать элементы управления. Для этого сначала необходимо добавить области просмотра с палитры *Презентация*. В данной модели будет четыре окна просмотра: главная страница (*viewAreaG*), 3D анимация (*viewArea3D*), статистика (*viewAreaS*) и процессы модели (*viewArea*) (Рисунок 154).

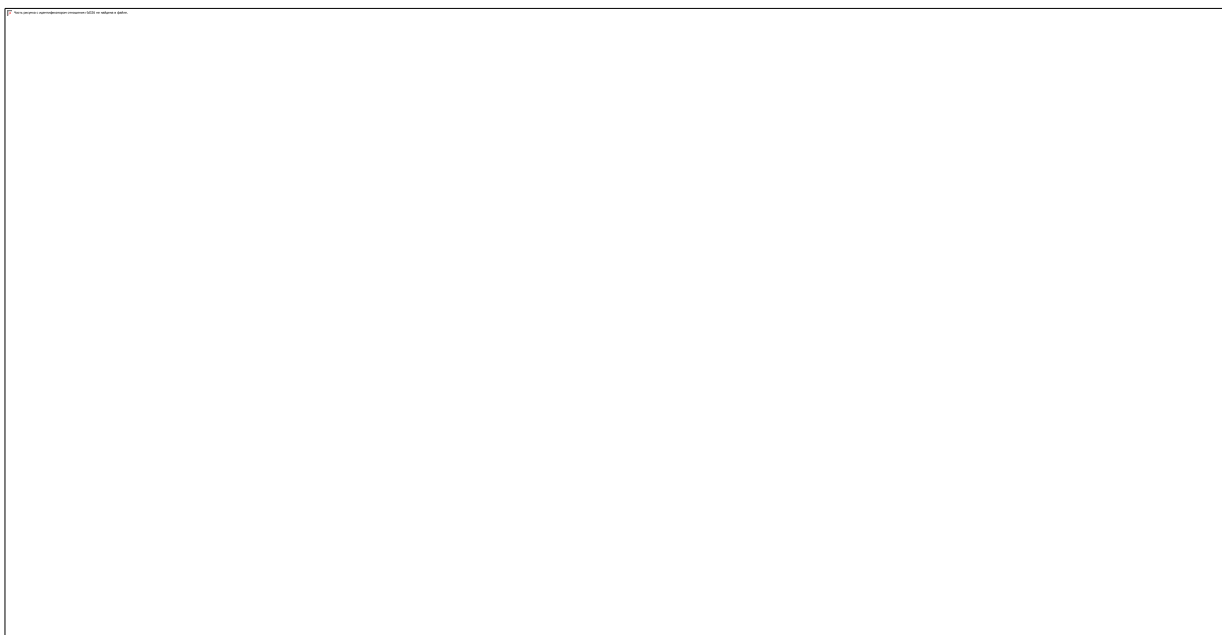



Рисунок 154 – Окна просмотра модели

Для правильного расчета количества дней уборки зерновых, необходимо добавить код, который будет их рассчитывать: `"day " + (int)(time()/day()+1)`.

Для осуществления между окнами необходимо в кнопки управления добавить ссылки в действия:

- 1) `viewAreaG.navigateTo();`
- 2) `viewArea3D.navigateTo();`
- `if (window3d.createCamera().equals(cameraTop)) {`
 `window3d.setCamera(camera, false, 1000);`
`}`
- 3) `viewAreaS.navigateTo();`
- 4) `viewArea.navigateTo();`

Настройки окна презентации настраиваются в свойствах эксперимента  Simulation, в секции Окно (Рисунок 155).

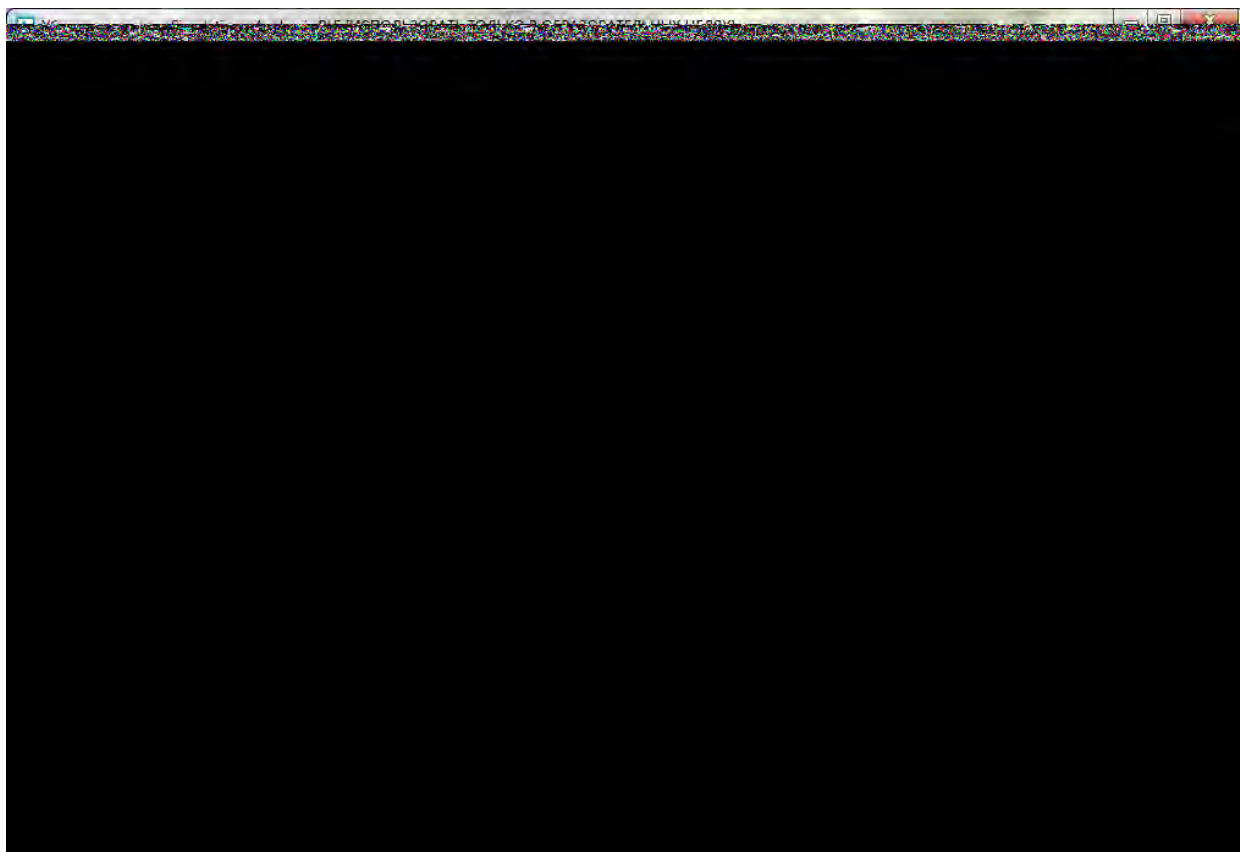


Рисунок 156 – Окно презентации модели

После завершения всех этапов разработки запускаем модель уборки зерновых культур (Рисунок 157).

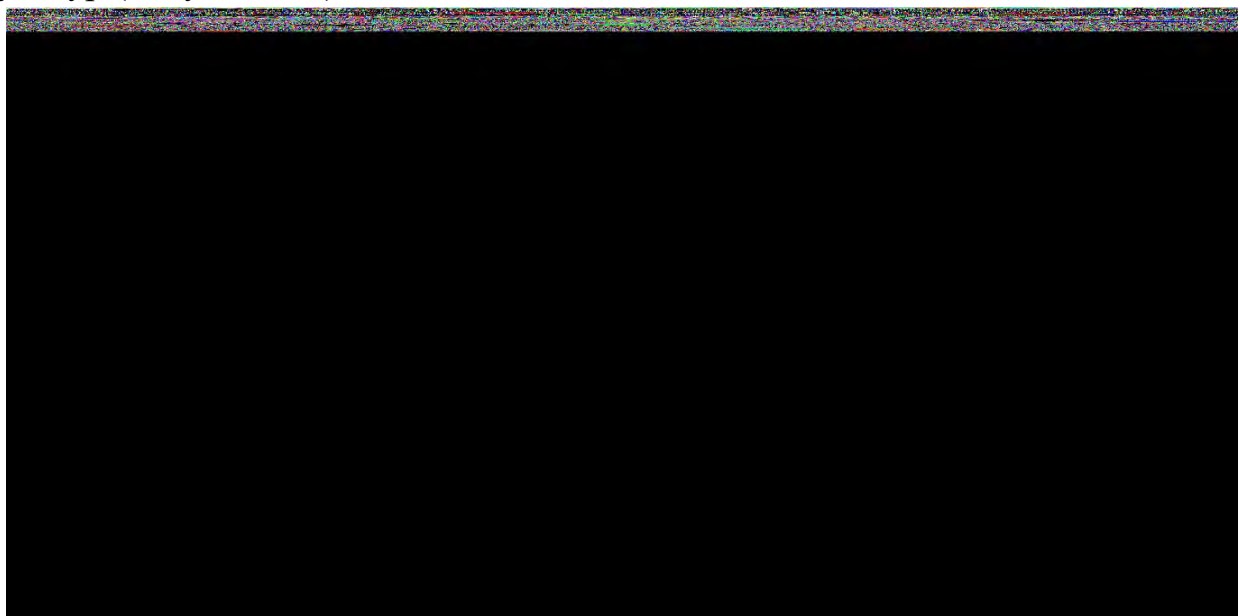


Рисунок 157 – Анимация прогона модели в AnyLogic, окно *Главная страница*

При прогоне модели на главной странице можно наблюдать процесс уборки зерна, следить за временем уборки. Здесь можно ускорить или замедлить прогон модели.

Для того чтобы наблюдать за процессом уборки зерновых культур в 3D формате необходимо перейти в 3D окно (Рисунок 158).

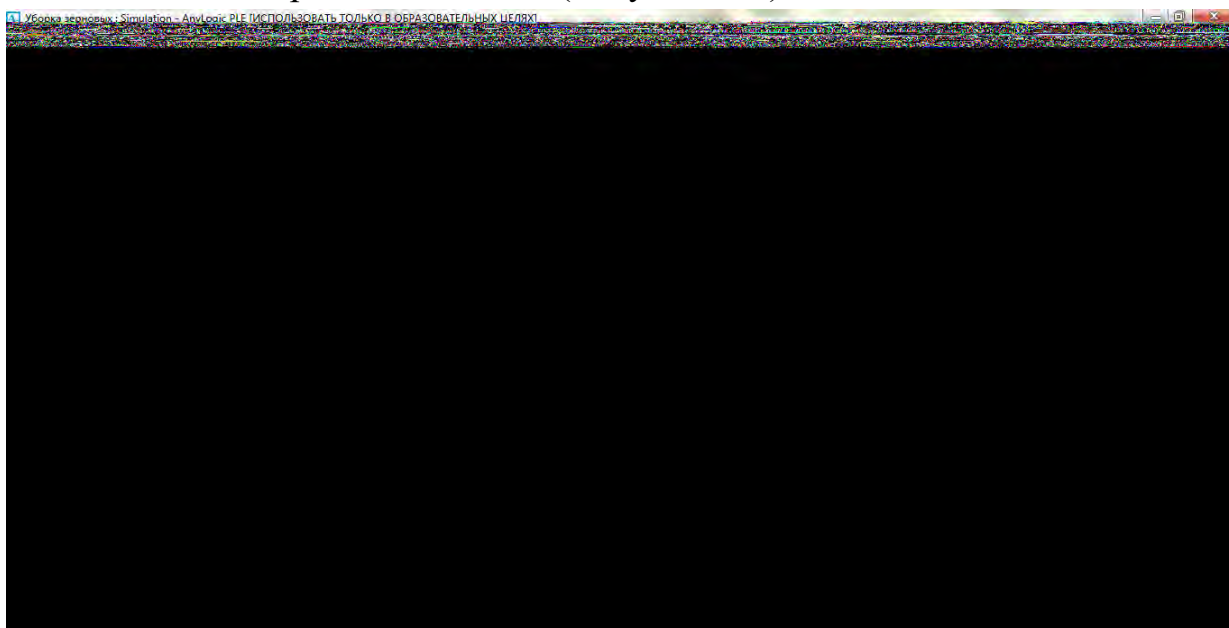


Рисунок 158 – Прогон модели в AnyLogic. *Окно 3D анимация*

Также можно посмотреть схемы процессов, происходящих при уборке, для этого необходимо нажать на кнопку Процессы модели (Рисунок 159).

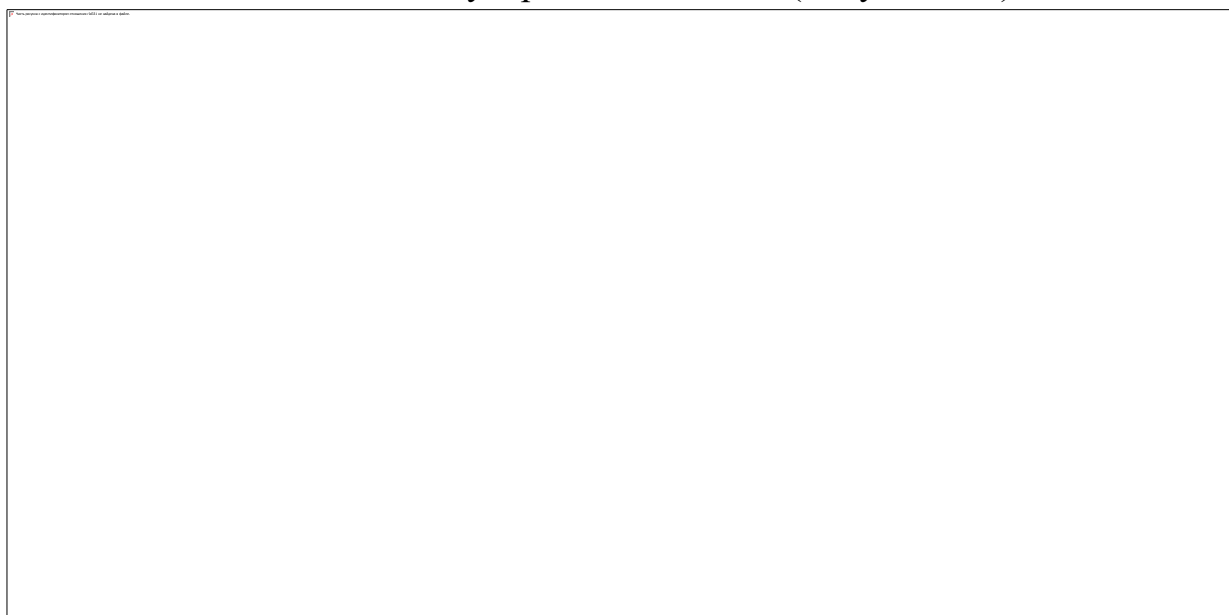


Рисунок 159 – Схемы процессов в модели в AnyLogic,
окно Процессы модели

Для просмотра статистики процессов необходимо перейти на вкладку *Статистика и показатели* (Рисунок 160).

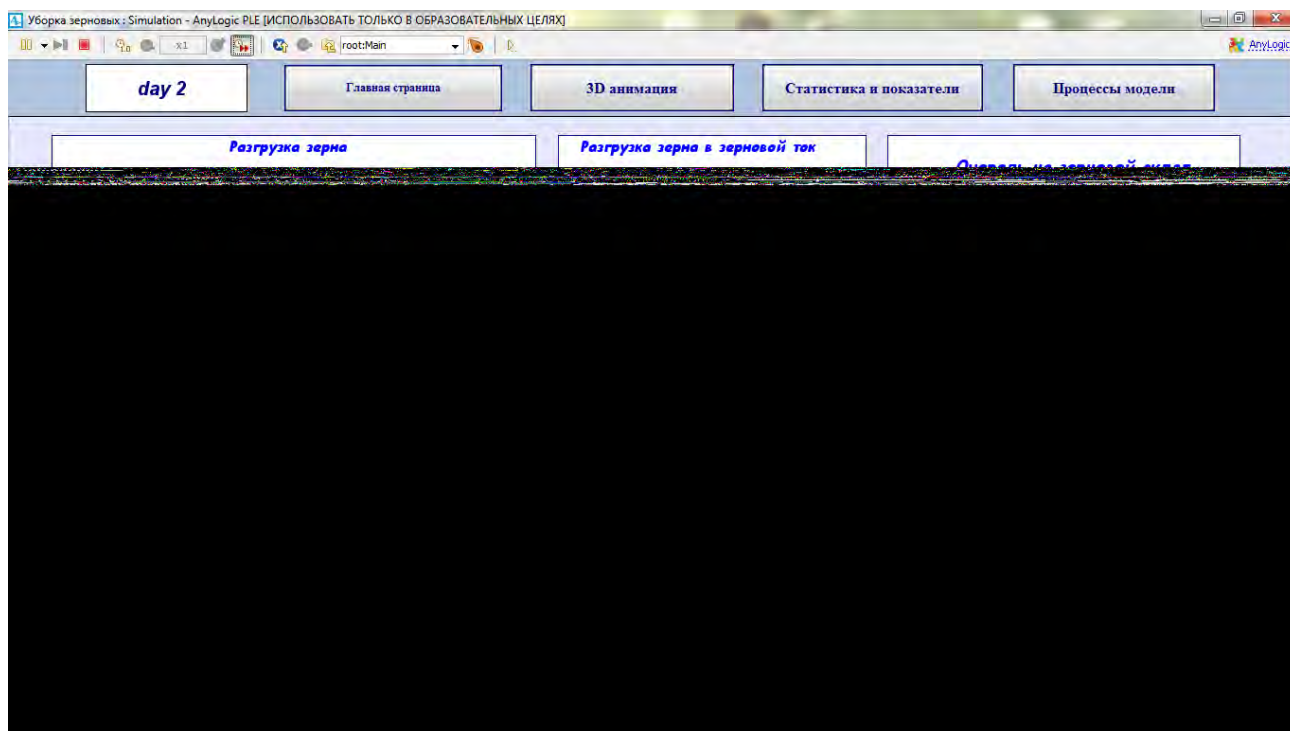


Рисунок 160 – Прогон модели в AnyLogic,
окно «Статистика и показатели»

На рисунке видно, что коэффициент загрузки автомобилей почти равен 1, то есть большую часть времени комбайны находятся в работе, однако, время ожидания машин комбайнами достаточно велико. Коэффициент загрузки пункта приема зерна не высокий, всего – 0,62, а также ожидание разгрузки достаточно длительное и составляет 0,6.

4.2.3. Проведение оптимизационных экспериментов на модели

После первого прогона модели срок уборки зерновых культур составил 31 день. Для полноценного анализа времени уборки зерновых культур необходимо, посмотреть, как будет изменяться срок уборки в зависимости от количества техники. Для этого необходимо сделать несколько прогонов модели с разным количеством техники. Вначале следует добавить еще один комбайн и снова прогнать модель (Рисунок 161).

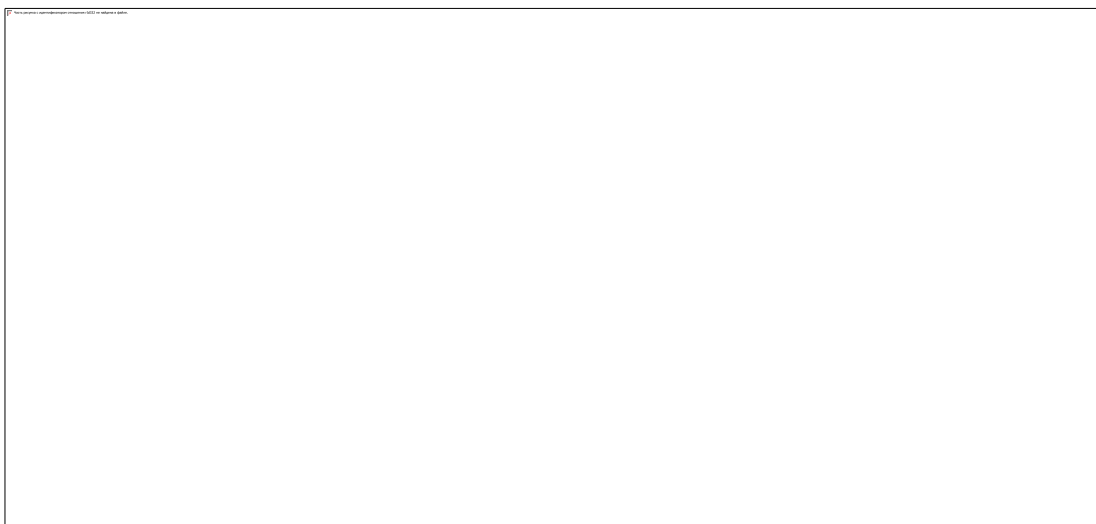


Рисунок 161 – Статистика и показатели прогона с 3 комбайнами

После окончания прогона срок уборки зерновых культур при использовании трех комбайнов составил 20 дней, а время ожидания равняется 0,5 и 0,4. Ожидание на складе не значительное.

Далее еще раз необходимо прогнать модель, но уже с четырьмя комбайнами и тремя машинами (Рисунок 162).

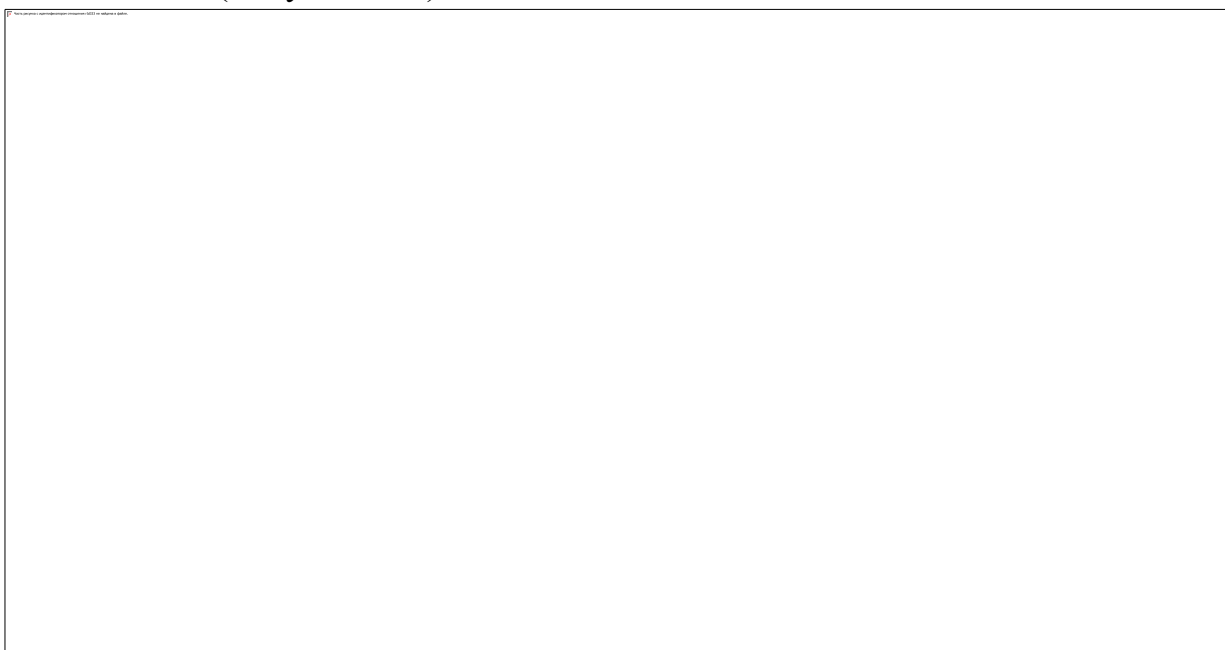


Рисунок 162 – Статистика и показатели прогона с 4 комбайнами

После окончания прогона срок уборки зерновых культур при использовании четырех комбайнов составил 13 дней. Показатели времени ожидания машин не превышало 0,4. Очередь на склад не значительная.

Далее снова необходимо добавить еще один комбайн и машину и прогнать модель (Рисунок 163).

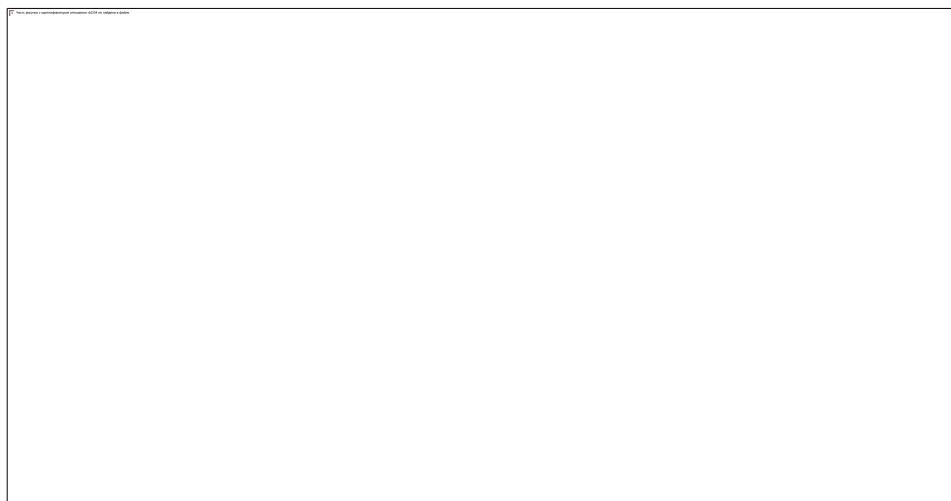


Рисунок 163 – Статистика и показатели прогона с 5 комбайнами

Уборка зерновых составила 9 дней, время ожидания машин в целом выше 0,4, очередь на зерновой склад возросла.

Прогон модели показал, что процесс уборки зерновых культур организован неэффективно. Сократить сроки уборки зерновых культур, повысить массу и цену зерна можно, увеличив количество уборочной техники. Имитационная модель позволяет провести эксперименты с моделью и получить характеристики бизнес-процесса при разном количестве техники. Проведение трех прогонов модели с разным количеством техники (соответственно 3, 4 и 5 комбайнов) показало следующие характеристики процесса (табл. 9).

Таблица 9 – Зависимость длительности уборки и урожайности от количества техники

Количество комбайнов	Длительность уборки	Урожайность
2 комбайна	31	16,0
3 комбайна	20	20,5
4 комбайна	13	28,59
5 комбайнов	9	31,2

Как видно из таблицы, при 5 комбайнах минимальна длительность процесса уборки зерна и максимальна урожайность.

Контрольные вопросы к главе IV.

1. Сущность агентного моделирования.
2. Отличия агентного моделирования от концепции системной динамики и дискретно-событийного.
3. Понятие агента. Характеристики агента.
4. Структура системы агентного моделирования.

5. Компоненты модели системной динамики.
6. Взаимодействие между объектами классов.
7. Области применения агентного моделирования.
8. Приведите примеры постановок задач агентного моделирования.
9. Опишите последовательность создания агентной модели.

ГЛАВА 5. ПРИМЕНЕНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ.

5.1. Имитационное моделирование и облачные технологии.

Имитационные исследования являются сложным по сути, и длительным по времени процессом, который требует множества вычислений, в том числе распределенных. Для проведения таких исследований создаются различные системы автоматизации имитационных исследований. Необходимость организации распределенных вычислений в имитационных исследованиях обуславливается высокой сложностью моделей и большим объемом исходных данных.

Современные информационные технологии позволяют перевести часть задач по разработке и верификации имитационных моделей, а также частично – экспериментов с ними, в облако. Облачные вычисления – это технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис. В настоящее время крупные вычислительные облака состоят из тысяч серверов, размещенных в центрах обработки данных (ЦОД). Они обеспечивают ресурсами десятки тысяч приложений, которые одновременно используют миллионы пользователей.

Термин «облачные вычисления» применим для любых сервисов, которые предоставляются через сеть Интернет. Облачные сервисы можно разделить на три основные категории:

- инфраструктура как сервис (Infrastructure as a Service, IaaS);
- платформа как сервис (Platform as a Service, PaaS);
- программное обеспечение как сервис (Software as a service, SaaS).

На этих трех составляющих строятся принципы работы облаков, по сути «замещающих» для пользователей их собственную информационную инфраструктуру, или конкретную программно-аппаратную платформу, или ПО. А именно, разница заключается в методе хранения и обработке данных. Весь процесс происходит на сервере в сети, которую называют «облачной технологией».

Облачные технологии – это различные аппаратные, программные средства, методологии и инструменты, которые предоставляются пользователю, как интернет-сервисы, для реализации своих целей, задач, проектов.

Выделяют следующие *типы облаков*: частное, публичное, гибридное, общественное.

Частное облако (англ. private cloud) – инфраструктура, предназначенная для использования одной организацией, включающей несколько потребителей (например, подразделений одной организации), возможно также клиентами и подрядчиками данной организации. Частное облако может находиться в собственности, управлении и эксплуатации как самой организации, так и третьей стороны

(или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

Публичное облако (англ. public cloud) – инфраструктура, предназначенная для свободного использования широкой публикой. Публичное облако может находиться в собственности, управлении и эксплуатации коммерческих, научных и правительственных организаций (или какой-либо их комбинации). Публичное облако физически существует в юрисдикции владельца – поставщика услуг.

Гибридное облако (англ. hybrid cloud) – это комбинация из двух или более различных облачных инфраструктур (частных, публичных или общественных), остающихся уникальными объектами, но связанных между собой стандартизованными или частными технологиями передачи данных и приложений (например, кратковременное использование ресурсов публичных облаков для балансировки нагрузки между облаками).

Общественное облако (англ. community cloud) – вид инфраструктуры, предназначенный для использования конкретным сообществом потребителей из организаций, имеющих общие задачи (например, миссии, требований безопасности, политики, и соответствия различным требованиям). Общественное облако может находиться в кооперативной (совместной) собственности, управлении и эксплуатации одной или более из организаций сообщества или третьей стороны (или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

К преимуществам облачных вычислений можно отнести следующие:

- доступ к личной информации с любого компьютера, подключённого к Интернету;
- возможность работать с информацией с разных устройств (ПК, планшеты, телефоны и т.п.);
- Веб-сервисы работают в браузере любых операционных систем;
- возможность одновременной работы с информацией несколькими пользователями;
- невозможность потери информации;
- оперативный доступ к свежей и обновлённой информации;
- возможность пользоваться последней версией программ;
- консолидация информации с другими пользователями;
- обмен информацией с людьми из любой точки мира.

Однако облачные технологии имеют определенные недостатки:

- для получения доступа к услугам «облака» необходимо постоянное соединение с Интернет;
- ограничения по программному обеспечению, и иногда невозможность настройки его под цели пользователя;

- сложность обеспечения конфиденциальности данных и безопасности;
- высокая стоимость оборудования;
- перспектива существенной монетизации ресурса.

Основные свойства облачных технологий

Национальный Институт стандартов и технологий NIST (США) в своем документе “The NIST Definition of Cloud Computing” определяет следующие характеристики облаков:

- *возможность в высокой степени автоматизированного самообслуживания системы* со стороны провайдера. У потребителя есть возможность получить доступ к предоставляемым вычислительным ресурсам в одностороннем порядке по мере потребности, автоматически, без необходимости взаимодействия с сотрудниками поставщика услуг.

- *Наличие системы Broad Network Access.*

- *Широкий сетевой доступ.* Предоставляемые вычислительные ресурсы доступны по сети через стандартные механизмы для различных платформ, тонких и толстых клиентов, мобильных телефонов, планшетов, ноутбуков, рабочих станций и т.п.

- *Объединение ресурсов в пулы.* Вычислительные ресурсы провайдера объединяются в пулы для обслуживания многих потребителей по многоарендной (multi-tenant) мо-дели. Примерами такого рода ресурсов могут быть системы хранения, вычислительные мощности, память, пропускная способность сети.

Мгновенная эластичность. Ресурсы могут быть легко выделены и освобождены, в некоторых случаях автоматически, для быстрого масштабирования соразмерно спросу. Для потребителя возможности предоставления ресурсов являются неограниченными, то есть они могут быть присвоены в любом количестве и в любое время.

Измеряемый сервис. Облачные системы автоматически управляют и оптимизируют ресурсы с помощью средств измерения, реализованных на разном уровне абстракции применительно для разного рода сервисов (например, управление внешней памятью, обработкой, полосой пропускания или активными пользовательскими сессиями). Использованные ресурсы можно отслеживать и контролировать, что обеспечивает прозрачность как для поставщика, так и для потребителя, использующего сервис.

Модели обслуживания облачных технологий. В настоящее время принято выделять три основные модели обслуживания облачных технологий, которые иногда называют «слоями облака». Эти три слоя – услуги инфраструктуры, услуги

платформы и услуги приложений – отражают строение не только облачных технологий, но и информационных технологий в целом.

К услугам инфраструктуры можно отнести набор физических ресурсов, таких как серверы, сетевое оборудование и накопители, предлагаемые заказчикам в качестве услуг. Услуги инфраструктуры решают задачу надлежащего оснащения центров обработки данных, предоставляя вычислительные мощности по мере необходимости. Примерами услуг инфраструктуры служат IBM SmartCloud Enterprise, VMWare, Amazon EC2, Windows Azure, Google Cloud Storage, Parallels Cloud Server и многие другие.

Услуги платформы – это модель обслуживания, в которой потребителю предоставляются приложения как набор услуг. В него входят, в частности, промежуточное ПО как услуги, обмен сообщениями как услуги, интеграция как услуги, информация как услуги, связь как услуги и т.д. Например, рабочее место как услуга (Workplace as a Service – WaaS) позволяет компании использовать облачные вычисления для организации рабочих мест своих сотрудников. Данные как услуга (Data as a Service – DaaS) предоставляют пользователю дисковое пространство, которое он может использовать для хранения больших объемов информации. Безопасность как услуга (Security as a Service – SaaS) дает возможность пользователям быстро разворачивать продукты, позволяющие обеспечить безопасное использование веб-технологий. Примерами услуг платформы служат IBM SmartCloud Application Services, Amazon Web Services, Windows Azure, Boomi, Cast Iron, Google App Engine и др.

Услуги приложений (Software as a Service – SaaS) предполагают доступ к приложениям как к сервису, то есть приложения провайдера запускаются в облаке и предоставляются пользователям по требованию как услуги. Приложения доступны посредством различных клиентских устройств или через интерфейсы тонких клиентов, например, такие, как веб-браузер, или веб-почта, или интерфейсы программ. Потребитель при этом не управляет базовой инфраструктурой облака, в том числе сетями, серверами, операционными системами. На конечном пользователе лежит ответственность только за сохранность параметров доступа (логинов, паролей и т.д.) и выполнение рекомендаций провайдера по безопасным настройкам приложений.

Примерами SaaS являются Gmail, Google Docs, Netflix, Photoshop.com, Acrobat.com, Intuit Quick-Books Online, IBM LotusLive, Unyte, Salesfor-ce.com, Sugar CRM и WebEx. Реализацией SaaS является и значительная часть растущего рынка мобильных приложений.

В связи с этим в развитии имитационного моделирования последних лет наблюдается тенденция переноса ряда функционала в облако, облачные технологии. Ресурсы интернета позволяют облегчить использование информации, её хра-

нение. Информация становится доступной многим пользователям одновременно. Доступность данных в облаке позволяет всесторонне анализировать большие объёмы информации. Использование облачных технологий в моделировании сложных социально-экономических систем снижает нагрузку на вычислительное оборудование, серверы, манфреймы.

На сегодняшний день в облако уходит лишь часть функционала. Если рассматривать жизненный цикл имитационной модели, то он состоит из следующих составляющих:

- разработка;
- отладка;
- верификация;
- валидация;
- дизайн экспериментов;
- дизайн интерфейса;
- выполнение имитационных экспериментов;
- анализ и экспорт результатов.

Разработка имитационной модели всегда связана с работой с большими объёмами данных, в которой участвуют разработчик и заказчик.

Анализ проведённый А.В. Борщёвым, руководителем фирмы-разработчика системы AnuLogic, показал, что средства разработки имитационной модели (редакторы и отладчики) в ближайшем будущем вряд ли перейдут в облако. А вот всё остальное или в облаке или стремится туда (Рисунок 164).

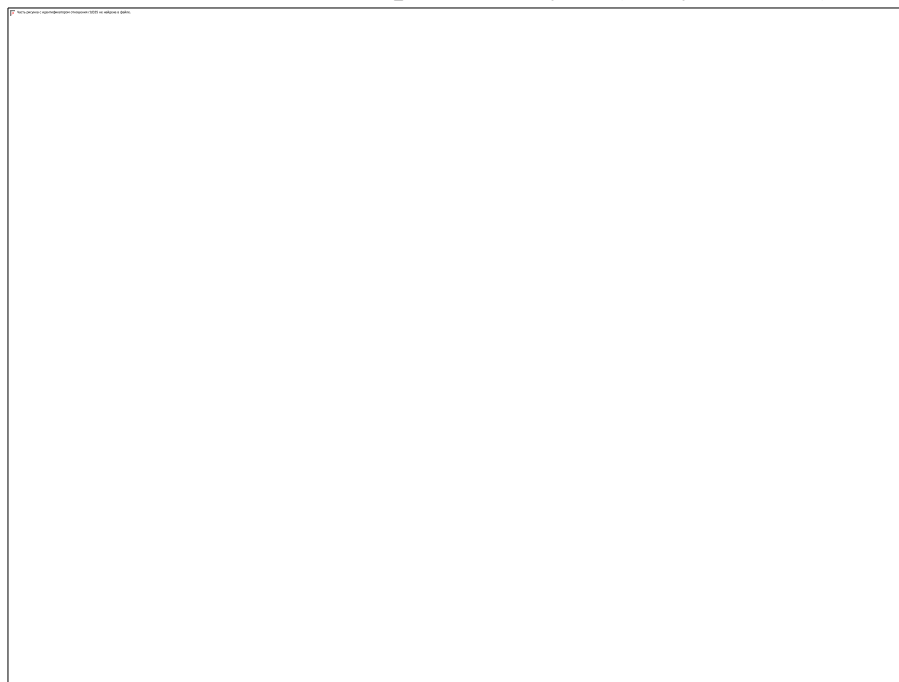


Рисунок 164 – Часть функционала при работе над имитационной моделью, переходящая в облако

Имитационные модели – вычислительно затратные объекты, и естественно использовать преимущества облака для повышения эффективности работы с ними. Во-первых, предоставляемый стандартным облачным сервисом узел является более мощным, чем локальный персональный компьютер. Во-вторых, его просто конфигурировать и менять. В-третьих, узел легко масштабируется под текущие потребности.

Облачные технологии позволяют осуществлять параллельные распределённые расчёты и вычисления. Использование множества параллельно работающих узлов, просто реализуемых, дает выигрыш по времени для многопрогонных экспериментов. При этом наибольший выигрыш от распараллеливания получается для экспериментов с абсолютно независимыми прогонами – то есть для всех кроме оптимизации. Оптимизация требует знания результатов предыдущих прогонов для запуска последующих. Запуская эксперимент в облаке, пользователь может выключить компьютер, а затем в любое время проверить статус и просмотреть результаты.

Ещё одно преимущество использования облачных расчётов в имитационном моделировании состоит в том, что загруженная в облако модель становится доступна многим пользователям, имеющим доступ в интернет.

При проведении имитационных исследований, рассматривая все этапы, прослеживаются следующие общие тенденции:

- интеграция исполнения этапов имитационного исследования в рамках одной программной системы. При этом создается система автоматизации имитационных исследований, охватывающая все этапы созданной имитационной модели;
- распределение функций исполнения отдельных этапов между различными пользователями и в результате создание многопользовательских систем. Например, в системах автоматизации имитационных исследований для корпоративных клиентов – одни отвечают за ввод исходных данных и их точность, другие за проведение экспериментов, третьи анализируют полученные результаты;
- обеспечение распределенных вычислений в рамках глобальных систем. Это – размещение некоторых сервисов, реализующих отдельные этапы исследования (например, моделирующее ядро) в сети Интернет;
- повышение простоты и доступности взаимодействия пользователя с системой на своем персональном компьютере, в основном за счет использования графики и анимации;
- создание локальных и глобальных баз данных моделей и существенное увеличение их размеров за счет детального мониторинга моделирования;
- 2D и 3D форм представления моделей и результатов.

Распределенное имитационное моделирование (distributed simulation) предполагает то, что выполнение одного прогона модели распараллеливается на не-

сколько процессоров, которые синхронизируются по модельному времени. Затраты на синхронизацию достаточно велики и зачастую не покрывают экономию от параллельной обработки. Поэтому множество параллельно работающих узлов в имитационном моделировании оправдано только при проведении экспериментов с моделью, требующих множества прогонов. Это такие эксперименты, как: Монте-Карло, варьирования параметров модели, оптимизационный.

При этом в облачном сервисе присутствует диспетчер по распределению прогонов по узлам и балансировщик нагрузки (Рисунок 165).

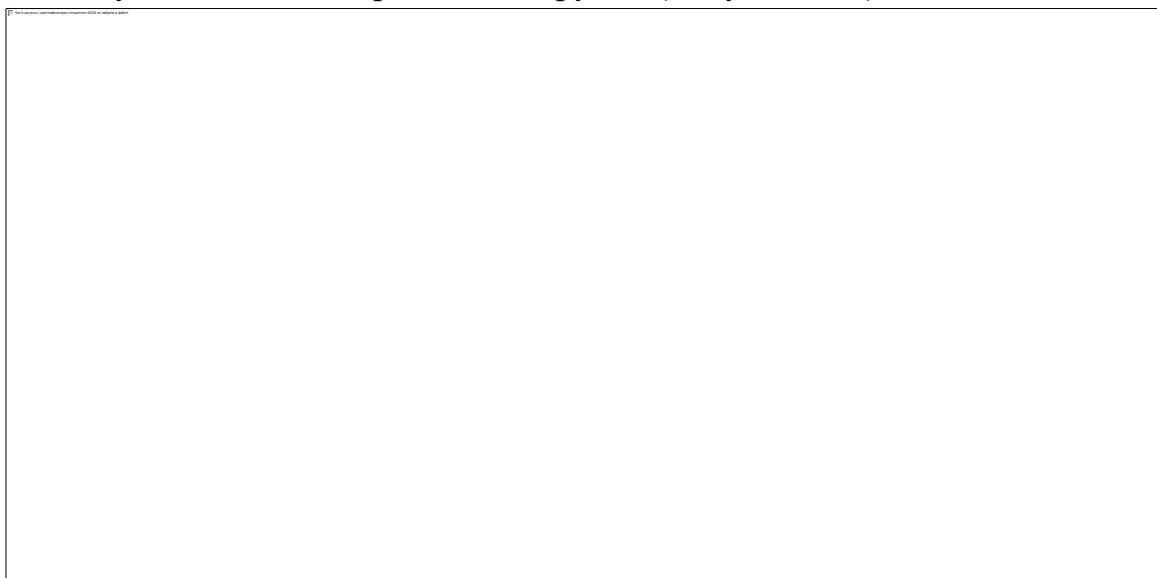


Рисунок 165 – Параллельное выполнение многопрогонных экспериментов с имитационной моделью в облаке

Если для разработки модели использование облачных сервисов не является достаточно эффективным, то для работы с уже готовой моделью облачные технологии удобны для реализации следующих задач:

- работа с версиями модели;
- задание и изменение входных параметров модели;
- наблюдение анимированного прогона и управление им;
- планирование и проведение сложных экспериментов с моделью;
- просмотр, анализ и экспорт результатов.

Это можно осуществлять с любого устройства (телефона, планшета и др.)

Облако в имитационном моделировании также эффективно может использоваться при доставке модели заказчику, который может ознакомиться с моделью, пройдя по высланной ему ссылке.

Значительные выгоды дает облако для коллективной работы с моделью. Результаты педенных прогонов модели, проведенные одними разработчиками/пользлвателями, хранятся в облаке и могут быть использованы другими лицами для статистической обработки результатов прогонов.

Также, использование моделей, находящихся в облаке в открытом доступе, позволит многочисленным исследователям – ученым, студентам и преподавателям, расширять свой кругозор относительно приемов моделирования и моделируемых процессов, дорабатывать уже созданные модели и создавать на их основе новые.

5.2. GPSS Cloud и AnyLogic Cloud для облачных имитационных исследований.

Тенденция дальнейшего развития имитационного моделирования ведёт к тому, что многие разработчики программного обеспечения в области имитационного моделирования создают сервисы, реализующие облачные технологии. Одним из таких продуктов является система GPSS Cloud. Так, проект построения облачной системы имитационного моделирования GPSS Cloud на основе языка GPSS World был реализован в компании ООО «Элина-Компьютер» (г. Казань).

Система GPSS Cloud опирается на применения распределенных вычислений в имитационном моделировании, в частности это разработанные и апробированные программные средства «Сервер GPSS» и «Системная шина моделирования». Сервер GPSS является программной реализацией сетевой системы имитационных исследований на основе GPSS World, а системная шина моделирования – шаблон построения системы автоматизированных имитационных исследований в рамках сервис-ориентированной архитектуры – одного из «столпов» облачных вычислений.

GPSS Cloud позволяет проводить удаленные комплексные имитационные исследования «по требованию». Исследователь со своего рабочего компьютера или планшета получает гибкий доступ к мощной базе математического, программного и аппаратного обеспечения, распространяемого в виде сервисов (услуг, служб) через стандартные интерфейсы.

Проект GPSS Cloud предоставляет возможности для проведения исследований по модели SaaS («программное обеспечение как услуга»), ориентированные на профессионалов имитационного моделирования. Данные возможности предоставляются как через Веб-интерфейс, так и в виде стандартизированных сервисов, что позволяет интегрировать функциональность этих сервисов в существующие и разрабатываемые приложения. Все это позволяет, как создавать системы автоматизации имитационных исследований для конечных пользователей, так и разрабатывать сторонний пользовательский интерфейс и клиентские приложения для различных платформ. Функциональность сервисов одинакова для любой предметной области и охватывает все основные этапы имитационных исследований, включая оптимизацию, планирование экспериментов и анализ результатов имитационного

моделирования. GPSS Cloud также предоставляет инструментарий для разработки и размещения Веб-приложений по модели PaaS. Этот веб-инструментарий позволяет разрабатывать собственные веб-интерфейсы и бизнеслогику, приспособленные для потребностей конкретного пользователя или группы пользователей. Примеры подобного сочетания моделей доставки облачных вычислений включают решения от Google (Google Apps – SaaS, Google App Engine – PaaS) и Salesforce.Com (salesforce.com – SaaS, force.com – PaaS). GPSS Cloud может быть размещена, как в собственном дата-центре, так и в открытом облаке по модели IaaS, например в AWS – сервисы Amazon EC2 (вычисления), Amazon S3 и Amazon EBS (хранилище).

GPSS Cloud по модели SaaS, как и все современные Веб-приложения, построена по модели трехуровневой архитектуры путем выделения трех логических уровней приложения – уровня представления, уровня бизнес-логики и уровня данных. Каждый уровень GPSS Cloud физически обособлен и занимает отдельную часть дата-центра.

Ядро приложения GPSS Cloud существует на уровне бизнес-логики и горизонтально масштабируется на множество виртуальных машин. Приложение построено по принципу «Multi-Tenancy», т.е. каждый экземпляр приложения не закреплен за конкретным пользователем, а обслуживает множество пользователей (1:M). Последнее достигается выделением всех пользовательских данных в отдельный кластер системы хранилища данных, что позволяет приложению не хранить данные о сеансах моделирования: запрос пользователя распределяется балансировщиком нагрузки на доступную виртуальную машину, а необходимые данные подгружаются из сервисов хранилища данных, и после окончания выполнения запроса соответствующие данные и состояние сохраняются обратно в хранилища данных. Поскольку все экземпляры приложения не содержат критичных данных и являются идентичными, дополнительные экземпляры приложения при масштабировании могут быть быстро развернуты из стандартного виртуального образа GMI (GPSS Cloud Machine Image). Уровень данных представляет собой сервиса хранилища данных, состоящую из кластера серверов ХД и мастер-сервера, который осуществляет распределение данных между узлами СХД. Кроме того мастер-сервер реализует стандартный интерфейс к СХД, инвариантный к ее программно-аппаратным особенностям. Данные в СХД хранятся следующих типов – данные о моделируемой предметной области, сценарии (модели) и результаты ИМ, а также предпочтения пользователя, используемые в частности для настроек интерфейса веб-приложения.

Уровень представления образует как доступный веб-интерфейс, так и различные сторонние клиентские приложения, использующие функциональность GPSS Cloud через предоставляемые стандартные интерфейсы. Стандартные ин-

терфейсы между уровнями реализовываются в виде общепринятых веб-сервисов (SOAP) и REST-сервисов (HTTP, XML). Кроме того, сам уровень бизнес-логики строится в соответствии с принципами сервис-ориентированной архитектуры (SOA).

Сервис-ориентированная архитектура (SOA) – это современный подход к разработке распределенных модульных приложений, основанный на выделении в качестве элементарных блоков приложения многократно-используемых автономных и обособленных IT-сервисов с четким интерфейсом, взаимодействующих между собой посредством обмена стандартизированными сообщениями через вычислительную сеть. Построенное на принципах SOA приложение является гетерогенным и распределенным в том смысле, что принципы внутреннего построения и надлежащей архитектуры отдельных сервисов являются несущественными, как и их физическое нахождение. Отметим, что, несмотря на то, что SOA часто реализуется именно в виде Веб-сервисов (и REST-сервисов), применение одних лишь Веб-сервисов не позволяет построить сервис-ориентированных решений, если не следовать принципам SOA.

Для интеграции сервисов в единое облачное приложение применяется шаблон сервисной шины предприятия ESB (Enterprise Service Bus). ESB – особый способ построения SOA-решений, при котором сервисы, составляющие приложение, развертываются вокруг сервисной шины, образующей среду для их взаимодействия.

Предлагаемый облачный подход к построению систем имитационного моделирования позволяет существенно упростить и унифицировать проведение имитационных исследований для конечных пользователей, образуя единую имитационную среду. Единая имитационная среда абстрагирует пользователя от особенностей выполнения имитационных исследований и является инвариантной по отношению к оконечному аппаратному и программному обеспечению. Отличительным достоинством единой имитационной среды являются минимальные требования к пользовательскому оборудованию – для проведения самых сложных имитационных исследований достаточно самого простого Интернет-планшета.

Единая среда GPSS Cloud поддерживает следующие типы моделирования:

- обычное моделирование – независимое выполнение ряда имитационных экспериментов для одной модели в режиме «как можно быстрее» с динамическим мониторингом данных – параллельное выполнение не связанных (изолированных) друг с другом моделей или серии экспериментов.
- высокопроизводительное моделирование – выполнение сложной модели с применением суперкомпьютерных технологий.

- распределенное моделирование – согласованное выполнение сложной модели, состоящей из составных моделей, сторонних служб и служб взаимодействия с пользователем в реальном масштабе времени.

По структуре GPSS Cloud включает в себя ряд служб, реализующих отдельные этапы имитационных исследований:

- обычное имитационное моделирование;
- распределенное имитационное моделирование;
- высокопроизводительное имитационное моделирование;
- планирование экспериментов;
- оптимизация;
- сложные расчеты;
- графические расчеты (2D, 3D, стерео-3D);
- анализ результатов.

Под распределенным моделированием понимается согласованное во времени выполнение моделей, сторонних IT-служб и взаимодействие с пользователем.

Служба распределенного имитационного моделирования включает в себя универсальные часы моделирования и выполняет две функции:

- согласование выполнения отдельных имитационных моделей;
- согласование выполнения сложной имитационной модели в реальном масштабе времени для взаимодействия с пользователем.

Под экспериментом понимается выполняющаяся реализация модели, которая:

- создается службой имитационного моделирования;
- является виртуальной службой;
- реализует универсальный интерфейс модели.

Универсальный интерфейс модели позволяет передавать данные в модель, управлять процессом моделирования, запрашивать данные из модели. Под сторонней службой понимается служба, реализующая *универсальный интерфейс модели*, и тем самым не отличимая от эксперимента, что позволяет интегрировать ее наравне с другими моделями GPSS в единую сложную имитационную модель. Сторонними службами могут быть модели, написанные на других языках, а также прочие программные и аппаратные средства, «обернутые» в SOA -службу. Под пользователем понимается сторонняя SOA-служба, взаимодействующая с пользователем для выполнения следующих функций:

- управление процессом имитационного моделирования (приостановление, возобновление, отмена) в реальном масштабе времени;
- изменение параметров моделирования «на лету»;
- получение данных о ходе имитационного моделирования в реальном масштабе времени.

Единая имитационная среда GPSS Cloud дает следующие возможности:

- реализация концепции SaaS («ПО как услуга»);
- расширение возможностей GPSS Cloud:
- синхронизация;
- удаленное управление процессом моделирования;
- передача данных в реальном масштабе времени.
- инвариантность к платформам и технологиям систем автоматизации имитационных исследований.

- инвариантность к местонахождению компонент систем автоматизации имитационных исследований.

- реализация высокопроизводительного имитационного моделирования.

- реализация распределенного имитационного моделирования с синхронизацией имитационных моделей.

Рассмотренная инфраструктура облачного моделирования на основе GPSS Cloud существенно сокращает время проведения имитационных исследований, а также делает его более доступным для широкого круга пользователей.

Система AnyLogic, работа в которой описана в главе IV, интегрируется с веб-порталом AnyLogic Cloud, который позволяет хранить, запускать модели и предоставлять к ним доступ. Загруженные в AnyLogic Cloud модели не требуют наличия программы AnyLogic на компьютере конечного пользователя. Они обрабатываются сервером и запускаются непосредственно в браузере. Для модели в облаке можно создавать собственные графические интерфейсы, встраивать их в HTML-страницы, а также интегрировать модель с BI-платформами – автоматизированными системами обработки и визуализации данных.

AnyLogic Cloud API работает на платформе Amazon Web Services и доступен для всех. В веб-сервисе можно запускать модели. Пользователи AnyLogic также могут загружать модели в AnyLogic Cloud и делиться ими.

AnyLogic Cloud API дает следующие возможности:

- позволяет расширять интерфейс облачных моделей с помощью собственных визуальных элементов, включая элементы ввода и вывода данных. Например, можно создать и настроить виджет, диаграмму или таблицу агрегирования результатов моделирования или добавить в модель логотип компании;

- встраивать интерфейс модели со всеми визуальными элементами на свой сайт, запускать с него модель, не заходя на сайт облака, а результаты получать в виде отчетов. По сравнению со встраиванием на сайт только окна анимации, API позволяет настраивать входные данные модели прямо на веб-странице и выгружать результаты экспериментов;

- интегрировать модель с BI-платформами, например Tableau или Power BI. Такие платформы обрабатывают и визуализируют результаты экспериментов для более удобного анализа данных;

Пользователи Private Cloud (коммерческая версия) имеют дополнительные преимущества:

- все данные хранятся на собственных серверах компании-пользователя. Организация контролирует общее хранилище моделей, аккаунты пользователей, результаты экспериментов и базу данных. Частное облако также можно настроить в соответствии с корпоративными требованиями безопасности;

- модели запускаются без временных ограничений и на вычислительных мощностях любого объёма;

- гибкость облака позволяет настраивать его под нужды компании и интегрировать его с коммерческой IT-инфраструктурой.

Независимо от языка, функции API переводит звонки на HTTP-запросы которые переходят на облачный сервер AnyLogic, ожидает и возвращает ответы. Типичная последовательность запросов/ответов HTTP представлена на Рисунке 166.

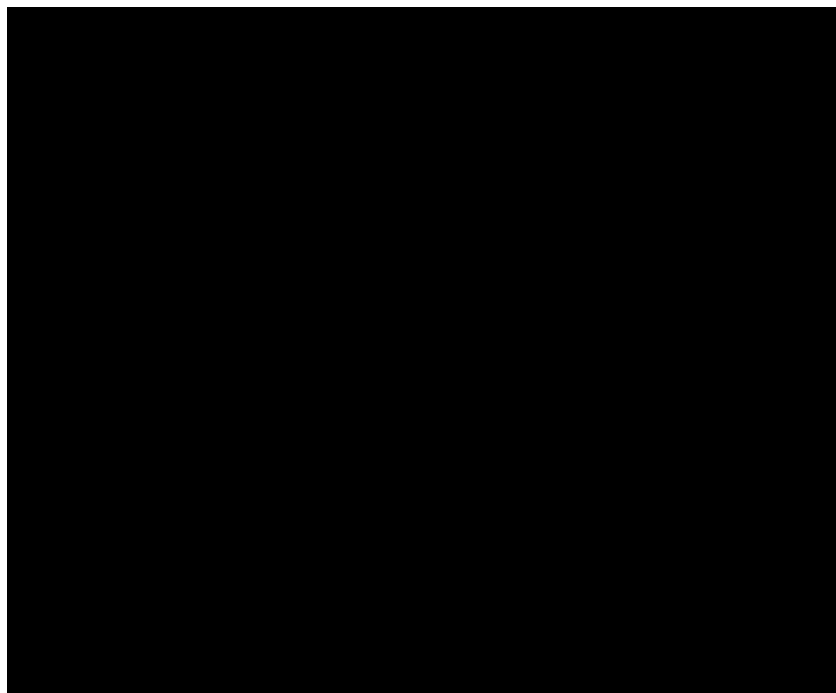


Рисунок 166 – Последовательность запросов/ответов HTTP

Итак, использование облачных сервисов в имитационном моделировании дает существенные преимущества разработчику/пользователю модели, которые наиболее ярко проявляются при работе пользователя с уже разработанной моделью при выполнении экспериментов с моделью, существенно сокращая время на проведение данной операции. Это дает возможность создания банка моделей, коллективного использования конкретной модели, возможность ее доработки и мо-

дификации. Значительно упрощается взаимодействие заказчика и разработчика модели.

Контрольные вопросы к главе V

1. Что такое облачные технологии, облачные вычисления?
2. На каких принципах строится работа облаков?
3. Какие типы облаков вы знаете? Охарактеризуйте их.
4. Перечислите преимущества облачных вычислений.
5. Каковы недостатки облачных технологий?
6. Основные свойства облачных технологий.
7. Функционал имитационного моделирования, уходящий в облако.
8. Что такое распределенное имитационное моделирование (distributed simulation).
9. Каковы возможности GPSS Cloud для облачных имитационных исследований?
10. Каковы возможности AnyLogic Cloud для облачных имитационных исследований?

Рекомендуемые источники:

1. Акопов, А.С. Имитационное моделирование: учебник и практикум для академического бакалавриата / А.С. Акопов. – М.: Издательство Юрайт, 2014. – 389 с.
2. Боев, В.Д. Компьютерное моделирование: Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7/ В.Д. Боев. — СПб.: ВАС, 2014. – 432 с.
3. Карпов, Ю.В. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5/ Ю.В. Карпов. – С.Пб.: БХВ-Петербург, 2006. – 400 с.
4. Каталевский, Д.Ю. Основы имитационного моделирования и системного анализа в управлении/ Д.Ю. Каталевский. – М.:Издательский дом «Дело» РАНХиГС, 2015. – 496 с.
5. Кораблев, Ю.А. Имитационное моделирование. Учебник / Ю.А. Кораблев. – Москва: КНОРУС, 2017. – 146 с.
6. Лычкина, Н.Н. Имитационное моделирование экономических процессов. Учебное пособие/ Н.Н. Лычкина. – М.: Инфра-М, 2012. – 253 С.
7. Митраков А.А. Основы проектирования системы агентного моделирования Пермский государственный национальный исследовательский университет, кафедра математического обеспечения вычислительных систем Пермь, Россия/ IV Международная студенческая электронная научная конференция «Студенческий научный форум», 15 февраля – 31 марта 2012 года. – Электронный ресурс. – Режим доступа - <https://www.rae.ru/forum2012/219/2702>.
8. Советов, Б.Я. Моделирование систем: учебник для академического бакалавриата / Б. Я. Советов, С. А. Яковлев. – 7-е изд. – М. : Издательство Юрайт, 2019. – 343 с.
9. Строгалева, В.П. Имитационное моделирование : учебное пособие / В. П. Строгалева, И. О. Толкачева. – 2-е изд., испр. и доп. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2015. – 295.
10. Агентные модели и Big Data: аналитика нового поколения. – Электронный ресурс. – Режим доступа <https://www.anylogic.ru/blog/agentnoe-modelirovanie-i-big-data-analitika-novogo-pokoleniya>.

Интернет-ресурсы

1. <https://www.anylogic.ru>
2. www.gpss.ru
3. www.vensim.com

Приложения

Приложение 1

1. Выражения модели Модель «Динамика прибыль предприятия – схема погашения долга»

(01) FINAL TIME = 5
Units: Month
The final time for the simulation.

(02) INITIAL TIME = 0
Units: Month
The initial time for the simulation.

(03) SAVEPER =
TIME STEP
Units: Month [0,?]
The frequency with which output is stored.

(04) TIME STEP = 1
Units: Month [0,?]
The time step for the simulation.

(05) Балансовая прибыль= INTEG (
Выручка-Общие затраты,
200000)
Units: Rub

(06) Время=
12
Units: Month

(07) Время для выплаты долга=
60
Units: Month

(08) Время на строительство=
10
Units: Month

(09) Выручка=
Количество*Цена
Units: Rub/Month

(10) Долг= INTEG (

Кредит-Основная выплата,
500000)

Units: Rub

(11) Затраты на оплату труда=
120000

Units: Rub

(12) Количество=
10000

Units: Shtuk

(13) Кредит=
Новые инвестиции+Часть финансового долга

Units: Rub/Month

(14) Материальные затраты=
140000

Units: Rub

(15) Налоги=
Налогооблагаемый доход*Налоговая ставка

Units: Rub

(16) Налоговая ставка=
0.2

Units: Procent

(17) Налогооблагаемый доход=
500000

Units: Rub

(18) Необходимые инвестиции=
500000

Units: Rub

(19) Новые инвестиции=
(Время+Время на строительство)/Необходимые инвестиции

Units: Rub/Month

(20) Общие затраты = A FUNCTION OF(Затраты на оплату труда
,Материальные затраты,Прочие затраты)

Общие затраты=

Материальные затраты+Затраты на амортизацию
+Затраты на оплату труда

+Затраты на социальные нужды+Прочие затраты
Units: Month

(21) Основная выплата=
Часть прибыли на погашение кредита/Время для выплаты долга
Units: Rub/Month

(22) Прочие затраты=
10000
Units: Rub

(23) Резервный фонд=
 $1/5 \times \text{Чистая прибыль}$
Units: Rub

(24) Фонд накопления=
 $1/7 \times \text{Чистая прибыль}$
Units: Rub

(25) Цена=
700
Units: Rub/Stuk

(26) Часть прибыли на погашение кредита=
 $\text{Чистая прибыль} - \text{Резервный фонд} - \text{Фонд накопления}$
Units: Rub

(27) Часть финансового долга=
500000
Units: Rub

(28) Чистая прибыль=
 $\text{Балансовая прибыль} - \text{Налоги}$
Units: Rub

Приложение 2

Уровни:

Имя элемента, единицы измерения	Выражение (значение)
Автомобили А, шт.	Производство автомобилей А-Продажи автомобиля А
Автомобили В, шт.	Производство автомобилей В-Продажи автомобиля В
Проданные автомобили А, шт.	Продажи автомобиля А

Проданные автомобили В, шт.	Продажи автомобиля А
--------------------------------	----------------------

Темпы:

<i>Имя элемента, единицы измерения</i>	<i>Выражение (значение)</i>
Продажи автомобиля А, авто в месяц	QUANTUM(Производство автомобилей А*Коэффициент продаж автомобилей А, 1)
Продажи автомобиля В, авто в месяц	QUANTUM(Производство автомобилей В*Коэффициент продаж автомобилей В, 1)
Производство автомо- билей А, авто в месяц	IF THEN ELSE(Выручка от продаж автомобилей А>500000,30+Time, 30)
Производство автомо- билей В, ед.	IF THEN ELSE(Выручка от продаж автомобилей В>800000,15+Time, 15)

Вспомогательные переменные:

<i>Имя элемента, единицы измерения</i>	<i>Выражение (значение)</i>
Выручка, руб./мес.	Выручка от продаж автомобилей А+Выручка от продаж автомобилей В
Выручка от продаж ав- томобилей А, руб./мес.	Проданные автомобили А*(Цена автомобиля А*Коммерческая надбавка-Затраты на производ- ство автомобилей А)
Выручка от продаж ав- томобилей В, руб./мес.	Проданные автомобили В*(Цена автомобиля В*Коммерческая надбавка-Затраты на производ- ство автомобилей В)
Затраты на производ- ство, руб./мес.	Амортизация+Фонд оплаты труда
Затраты на производ- ство автомобилей А, руб.	Необходимое количество металла для производ- ства автомобиля А*Цена на единицу металла
Затраты на производ- ство автомобилей В, руб.	Необходимое количество металла для производ- ства автомобиля В*Цена на единицу металла
Затраты на хранение, руб.	Производство автомобилей А*Затраты на хране- ние автомобиля А+Производство автомобилей В *Затраты на хранение автомобиля В
Коэффициент продаж автомобилей А, dmn1.	SMOOTH(IF THEN ELSE(Цена автомобиля А>1e+006, 0.4 , 1-Цена автомобиля А*1e-006) , 20)
Коэффициент продаж автомобилей В, dmn1.	IF THEN ELSE(Цена автомобиля В>1e+006, 0.2 , 1-Цена автомобиля В*1e-006)
Необходимое количе- ство металла для про- изводства автомобиля А, ед.	IF THEN ELSE(Качество металла=1, 30 , IF THEN ELSE(Качество металла=2, 20 , IF THEN ELSE(Качество металла=3, 10 , 0)))

Необходимое количество металла для производства автомобиля В, ед.	IF THEN ELSE(Качество металла=1, 45 , IF THEN ELSE(Качество металла=2, 30 , IF THEN ELSE(Качество металла=3, 15 , 0)))
Общие затраты, руб.	Затраты на производство+Затраты на хранение
Расходы на реализацию, руб./мес.	Расходы на продажу в автосалонах+Расходы на транспортировку автомобилей в автосалоны
Расходы на транспортировку автомобилей в автосалоны, руб./авто.	Производство автомобилей $A*2000 + \text{Производство автомобилей } B*3500$
Рентабельность, dmn1	QUANTUM((Чистая прибыль/(Затраты на производство автомобилей А+Затраты на производство автомобилей В))*100 , 1)
Фонд оплаты труда, руб.	Средние затраты на оплату труда на 1 работника*Численность сотрудников
Цена на единицу металла, руб.	IF THEN ELSE(Качество металла=1, 20000 , IF THEN ELSE(Качество металла=2, 40000 , IF THEN ELSE(Качество металла=3, 50000 , 0)))
Чистая прибыль, руб.	Выручка-Общие затраты-Расходы на реализацию

Константы:

<i>Имя элемента, единицы измерения</i>	<i>Значение</i>
Прочие, руб./мес.	50000
Затраты на хранение автомобиля А, руб/авто.	10000
Затраты на хранение автомобиля В, руб/авто.	20000
Качество металла, dmn1	1
Коммерческая надбавка, dmn1	1.15
Расходы на продажу в автосалонах, руб/мес.	100000
Средние затраты на оплату труда на 1 работника, руб.	50000
Цена автомобиля А, руб.	500000
Цена автомобиля В, руб.	800000
Численность сотрудников, чел.	100

Приложение 3

Модель «Стратегии кормопроизводства
сельскохозяйственного предприятия»

- (01) FINAL TIME = 12
Units: Month
- (02) INITIAL TIME = 1
Units: Month
- (03) SAVEPER = TIME STEP
Units: Month [0,12]
- (04) TIME STEP = 1
Units: Month [0,12]
- (05) Выбраковка скота = Поголовье молочного стада*Процент выбраковки
Units: гол/Month
- (06) Выручка = Надой на одну корову * Поголовье молочного стада
*Цена молока + Стоимость проданной говядины
Units: руб/Month
- (07) Выход телят на одну голову скота = 0.08
Units: гол/Month/гол
- (08) Жирность молока = 3.5+Содержание углеводов * 0.4
Units: кг/центнер
- (09) Качество кормов = 1
Units: кг/центнер
- (10) Молодняк = Выход телят на одну голову скота*Поголовье молочного стада
Units: гол/Month
- (11) Надой на одну корову = IF THEN ELSE((Качество кормов=1) ,300,280)
Units: литр/гол/Month
- (12) Падеж скота = Процент падежа*Поголовье молочного стада
Units: гол/Month
- (13) Поголовье молочного стада = INTEG (Молодняк - Выбраковка скота - Падеж скота, 500)
Units: гол
- (14) Прибыль = Выручка – Себестоимость - Убыток от падежа
Units: руб/Month
- (15) Процент выбраковки = 0.01
Units: гол/Month/гол
- (16) Процент падежа = 0.04
Units: гол/Month/гол
- (17) рентабельность = Прибыль/Себестоимость
Units: Dmnl
- (18) Себестоимость = Поголовье молочного стада*(Стоимость содержания одной головы скота (без учета затрат на корм) +Стоимость кормов на одну голову скота в месяц)
Units: руб/Month
- (19) Содержание углеводов = IF THEN ELSE(Качество кормов=1, 3.2 , 2.5)
Units: кг/центнер

(20)	Стоимость кормов на одну голову скота в месяц = IF THEN ELSE(Качество кормов=1, 4800 , (IF THEN ELSE(Качество кормов=2, 3700 , 2800)))
	Units: руб/(Month*гол)
(21)	Стоимость одной головы скота (на мясо) = 10000
	Units: руб/гол
(22)	Стоимость проданной говядины = Выбраковка скота*Стоимость одной головы скота (на мясо)
	Units: руб/Month
(23)	Стоимость содержания одной головы скота (без учета затрат на корм) = 1500-STEP(200 , 4)+STEP(300, 9)
	Units: руб/гол/Month
(24)	Убыток от падежа = Падеж скота*Стоимость одной головы скота (на мясо)
	Units: руб/Month
(25)	Цена молока = IF THEN ELSE((Жирность молока>=4.6) ,7, 5)
	Units: руб/литр

Приложение 4 Модель

- (01) *Срок полезного использования* = 10
Units: год
- (02) *FINAL TIME* = 3
Units: год
- (03) *INITIAL TIME* = 0
Units: год
- (04) *SAVEPER* = *TIME STEP*
Units: год [0,?]
- (05) *TIME STEP* = 0.015625
Units: год [0,?]
- (06) *Амортизация* = *Основной капитал*/*Срок полезного использования*
Units: \$/год
- (07) *Валовой доход* = *Реализованная продукция*
Units: \$/год
- (08) *Время выставление счета* = 0.04
Units: год
- (09) *Время погашения* = 3
Units: год
- (10) *Время строительства* = 2
Units: год
- (11) *Всего инвестиции* = 200000
Units: \$

(12) *Выплата основной суммы = Выплата процентов/Время погашения*
Units: \$/год

(13) *Выплата процентов = Долг*"Ставка %"*
Units: \$/год

(14) *Дебиторская задолженность = INTEG (Реализованная продукция - Денежные поступления - Потери, Реализованная продукция /(1/Среднее время поступления денег + Доля потерь))*
Units: \$

(15) *Денежные поступления = Дебиторская задолженность/Среднее время поступления денег*
Units: \$/год

(16) *Долг= INTEG (Заемные средства-Выплата основной суммы, 0)*
Units: \$

(17) *Доля заемных средств = 0.6*
Units: Dmnl

(18) *Доля постоянных затрат = 0.6*
Units: \$/tonn

(19) *Доля потерь = 0.06*
Units: 1/год

(20) *Заемные средства = Новые инвестиции*Доля заемных средств*
Units: \$/год

(21) *Налог = Прибыль до налогообложения*Ставка процента*
Units: \$/год

(22) *Новые инвестиции=*
IF THEN ELSE(Time≥Время строительства, 0 , Всего инвестиции
/Время строительства)
Units: \$/год

(23) *Объем производства = Полезная мощность*
Units: tonn/год

(24) *Основной капитал= INTEG (Новые инвестиции-Амортизация, 0)*
Units: \$

(25) *Полезная мощность = IF THEN ELSE(Time>=Время строительства, Производственная мощность, 0)*
Units: tonn/год

(26) *Постоянные издержки = Объем производства*Доля постоянных затрат*
Units: \$/год

(27) *Потери = Дебиторская задолженность*Доля потерь*
Units: \$/год

(28) *Прибыль до налогообложения = Валовый доход-Постоянные издержки-Амортизация - Потери*
Units: \$/год

(29) *"Продукция, ожидающая реализации"= INTEG (Произведенная для реализации продукция - Реализованная продукция, Произведенная для реализации продукция *Время выставление счета)*
Units: \$

- (30) Произведенная для реализации продукция = Объем производства*Цена
Units: \$
- (31) Производственная мощность = 3000
Units: tonn/год
- (32) Реализованная продукция = "Продукция, ожидающая реализации"/Время выставление счета
Units: \$/год
- (33) Среднее время поступления денег = 0.09
Units: год
- (34) "Ставка %" = 0.12
Units: 1/год
- (35) Ставка процента = 0.4
Units: Dmnl
- (36) Цена = 770
Units: \$/tonn
- (37) Чистая прибыль = Прибыль до налогообложения - Налог - Выплата процентов
Units: \$/год
- (38) Чистый денежный поток = Денежные поступления + Заемные средства - Выплата процентов -Налог - Новые инвестиции -Постоянные издержки-Выплата основной суммы
Units: \$/год

Приложение 5

Выражения модели «Планирование регионального объема валового производства»

- (01) *FINAL TIME* = 100
Units: Year
The final time for the simulation.
- (02) *INITIAL TIME* = 0
Units: Year
The initial time for the simulation.
- (03) *SAVEPER* =
TIME STEP
Units: Year [0,?]
The frequency with which output is stored.
- (04) *TIME STEP* = 1
Units: Year [0,?]
The time step for the simulation.
- (05) Амортизация=Основные фонды*Норма амортизации основных фондов

- (06) Валовая продукция сельского хозяйства = INTEG (+Производство-Амортизация-Производственные затраты
-Налог на продукт-Прибыль-Плата за труд, 2.85042e+007)
- (07) ВРП на душу населения=Валовая продукция сельского хозяйства/Население
- (08) "Доход от с.-х. производства на душу населения"
="Доходы населения от с.-х. производства"/Население
- (09) "Доходы населения от с.-х. производства"=ФЗП-Подходный налог+Социальный налог
- (10) Заемные средства=Займы в год
- (11) Займы в год=1e+007
- (12) "Инвестиции (из прибыли)"= IF THEN ELSE(Финансовые результаты<=0 , 0 , Финансовые результаты *Норма накопления)
- (13) Инвестиции из бюджета= Инвестиции в сельское хозяйство региона
- (14) Инвестиции из федерального бюджета=200000
- (15) Инвестиции в сельское хозяйство региона=60000
- (16) Коэффициент износа основных фондов= 0.1
- (17) Коэффициент эластичности основных фондов=0.9
- (18) Коэффициент эластичности трудовых ресурсов=0.3
- (19) Налог на имущество с ЮЛ= Основные фонды*Ставка налога на имущество
- (20) Налог на прибыль=IF THEN ELSE(Прибыль>=0, Прибыль*Ставка налога на прибыль , 0)
- (21) Налог на продукт=Средняя ставка налогов*Валовая продукция сельского хозяйства
- (22) Население=500
- (23) Непроизводственное потребление=
IF THEN ELSE(Финансовые результаты>=0, Финансовые результаты

- "Инвестиции (из прибыли)" - Налог на прибыль , 0)

(24) Норма амортизации основных фондов = 0.1

(25) Норма накопления = 0.1

(26) Основные фонды = $INTEG (\text{Амортизация} + \text{Заемные средства} + \text{"Инвестиции (из прибыли)"} + \text{Инвестиции из бюджета} + \text{Инвестиции из федерального бюджета} - \text{Налог на имущество с ЮЛ} - \text{Потребление ОФ}, 1.25643e+008)$

(27) Плата за труд = $\text{Численность занятых} * \text{Средняя заработная плата по региону} * (1 - \text{Ставка социального налога})$

(28) Подоходный налог = $1.15648e+006$

(29) Потребление ОФ = $\text{Основные фонды} * \text{Коэффициент износа основных фондов}$

(30) Прибыль = $\text{Валовая продукция сельского хозяйства} - \text{Амортизация} - \text{Производственные затраты} - \text{Плата за труд} - \text{Налог на продукт}$

(31) Производственные затраты = $\text{Удельные материальные затраты} * \text{Валовая продукция сельского хозяйства}$

(32) Производство = $\text{"Темп научно-технического прогресса"} * \text{Численность занятых} \wedge \text{Коэффициент эластичности трудовых ресурсов} * \text{Основные фонды} \wedge \text{Коэффициент эластичности основных фондов}$

(33) Социальный налог = $2.31296e+006$

(34) Спрос на рабочую силу = $\text{Основные фонды} / \text{Средняя фондовооруженность}$

(35) Средняя заработная плата по региону = 32000

(36) Средняя ставка налогов = 0.3

(37) Средняя фондовооруженность = 452000

(38) Ставка налога на имущество = 0.02

(39) Ставка налога на прибыль = 0.2

(40) Ставка социального налога = 0.26

(41) *"Темп научно-технического прогресса"* = 0.5

(42) *Удельные материальные затраты* = 0.4

(43) *ФЗП* = *Плата за труд*

(44) *Финансовые результаты* = *Прибыль*

(45) *Численность занятых* = 278

Результаты прогона имитационной модели «Прогнозирование объемов
регионального валового производства»

