Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»

К.А. Аксенов, Н.В. Гончарова

ГИБРИДНОЕ МОДЕЛИРОВАНИЕ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ

Монография

Москва
2019

УДК 004.94
ББК 32.81
А42

Рецензенты:

Бегунов Н.А. – кандидат технических наук, доцент, менеджер по машинному обучению и большим данным, ПАО «МегаФон», г. Екатеринбург.

Круглов А.В. – кандидат технических наук, доцент лаборатории Промышленной разработки ПО, Университет Иннополис, г. Иннополис.

Аксенов К.А., Гончарова Н.В.

А42 Гибридное моделирование мультиагентных процессов преобразования ресурсов: монография / К.А. Аксенов, Н.В. Гончарова. – М.: Издательский дом Академии Естествознания, 2019. – 222 с.

ISBN 978-5-91327-606-3

Книга содержит описание основных проблем динамического моделирования мультиагентных процессов преобразования ресурсов. Разработанные модели и методы могут быть использованы в области производства, технологических и бизнес-процессов для создания систем поддержки принятия решений. Данная книга рекомендуется студентам и аспирантам технических вузов, прикладным программистам, аналитикам, руководителям информационных отделов и фирм для практического изучения и реализации методов и алгоритмов программирования ситуационного, имитационного, экспертного и мультиагентного моделирования бизнес-процессов.

Библиогр.: 181 назв. Табл. 16. Рис. 101. Прил. 6.

ISBN 978-5-91327-606-3

© Аксенов К.А., Гончарова Н.В., 2019
© ИД «Академия Естествознания»
© АНО «Академия Естествознания»

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 5

1. ОБЗОР МЕТОДОВ И СРЕДСТВ СИТУАЦИОННОГО МОДЕЛИРОВАНИЯ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ 8
   - 1.1. Ситуационное управление организационно-техническими системами 8
     - 1.1.1. Ситуационный подход в управлении 8
     - 1.1.2. Рассмотрение организационно-технических систем с точки зрения процессов преобразования ресурсов 10
   - 1.2. Процесс принятия решений и информационные технологии в организационно-технических системах управления 16
     - 1.2.1. Системы поддержки принятия решений (СППР) 18
     - 1.2.2. СППР в стратегическом управлении 20
   - 1.3. Системы ситуационного моделирования 21
   - 1.4. Методы моделирования 24
     - 1.4.1. Имитационное моделирование 24
     - 1.4.2. Экспертное моделирование 26
     - 1.4.3. Ситуационное моделирование 28
     - 1.4.4. Мультиагентный подход 29
   - 1.5. Обзор и сравнение систем динамического моделирования ситуаций (СДМС) 32
     - 1.5.1. Обзор существующих СДМС 32
     - 1.5.2. Требования к СДМС 34
     - 1.5.3. Сравнительный анализ СДМС 35
   - Выводы 37
2. СИТУАЦИОННАЯ МОДЕЛЬ МУЛЬТИАГЕНТНОГО ПРОЦЕССА ПРЕОБРАЗОВАНИЯ РЕСУРСОВ 38
   - 2.1. Требования к модели 38
   - 2.2. Основные объекты ситуационной модели процессов преобразования ресурсов, их поведения и отношения 38
     - 2.2.1. Анализ работы классической ситуационной (семиотической) модели 59
     - 2.2.2. Анализ SIE-модели А.Ю. Филипповича 61
     - Выводы по разделу 2.2 64
   - 2.3. Анализ и выбор модели интеллектуального агента 65
   - 2.4. Анализ и выбор моделей представления знаний 68
     - 2.4.1. Анализ фреймовых моделей 70
     - 2.4.2. Применение фреймового подхода Швецова для построения концептуальной модели предметной области (КМПО) 71
   - 2.5. Алгоритмы работы системы (машина вывода) 72
     - 2.5.1. Алгоритм работы интеллектуального агента 73
     - 2.5.2. Алгоритм ситуационно-имитационного моделирования 77
   - Выводы 84

3. ОПИСАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ СИТУАЦИЙ BPSIM2 85
3.1. Функции системы 85
3.2. Принципы построения СДМС процессов преобразования ресурсов 86
3.2.1. Применение объектно-ориентированного подхода при создании системы 86
3.2.2. Математический аппарат 86
3.2.3. Информационное обеспечение 90
3.2.4. Программное обеспечение 90
3.2.5. Алгоритмическое обеспечение 90
3.2.6. Требования к аппаратному и программному обеспечению 92
3.3. Технология работы с системой 92
3.3.1. Основные этапы работы с СДМС BPsim2 92
3.3.2. Реализация механизма логического вывода на основе языка Transact-SQL 93
Выводы по разделу 3.3.2: 101
3.3.3. Описание фреймовой оболочки экспертных систем «Конструктор фрейм-систем» 101
3.3.4. Создание ситуационной моделипроцессов преобразования ресурсов 103
3.3.5. Проведение экспериментов с моделью 112
Выводы 115
4. ПРИМЕНЕНИЕ СИСТЕМЫ 116
4.1. Базовое предприятие 116
4.2. Моделирование ЗАО «Уральская индустриальная группа» 125
4.2.1. Процесс производства 126
4.2.2. Процесс монтажа 129
4.2.3. Модели агентов 129
Выводы 153
ЗАКЛЮЧЕНИЕ 154
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 156
СПИСОК ЛИТЕРАТУРЫ 158
ПРИЛОЖЕНИЯ. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СДМС 176
Приложение 1. Выбор критериев сравнения 176
Приложение 2. Система имитационного моделирования AnyLogic5.2 186
Приложение 3. Экспертная система реального времени G2 196
Приложение 4. Система моделирования и проектирования ARISToolSet 206
Приложение 5. СДМС BPsim2 211
Приложение 6. Результаты сравнения систем 220

ВВЕДЕНИЕ

Книга посвящена вопросам поддержки принятия решений (ППР). В большинстве случаев ППР заключается в генерации возможных альтернатив решений, их оценке и выборе лучшей альтернативы. При выборе альтернатив приходится учитывать большое число противоречивых требований и, следовательно, оценивать варианты решений по многим критериям.

Формализация методов принятия решений, их оценка и согласование являются чрезвычайно сложной задачей. Увеличение объема информации, поступающей в органы управления и непосредственно к руководителям, усложнение решаемых задач, необходимость учета большого числа взаимосвязанных факторов и быстро меняющаяся обстановка настоятельно требуют использовать вычислительную технику в процессе принятия решений. Характерной особенностью существующих в настоящее время систем поддержки принятия решений (СППР) является их ориентированность на узкий круг решаемых задач, что создает значительные трудности для лиц, принимающих решения (ЛПР).

В данной книге рассматриваются вопросы моделирования мультиагентных процессов преобразования ресурсов. К ним относятся большинство окружающих нас процессов: процессы, протекающие в производстве, технике, организационно-технических системах, экономике, окружающей среде. В настоящее время наблюдается существенный интерес к области мультиагентных систем, спецификой которых является наличие сообществ взаимодействующих агентов, отождествляющихся с ЛПР.

У истоков мультиагентного подхода лежат методы экспертного, имитационного и ситуационного моделирования. Существенный вклад в развитие данного направления внесли следующие ученые: Борщёв А.В., Вавилов А.А., Гольдштейн С.Л., Емельянов С.В., Исидзука М., Карпов Ю.Г., Клыков Ю.И., Попов Э.В., Поспелов Д.А.,

Прицкер А., Советов Б.Я., Форрестер Дж., Филиппович А.Ю., Чистов В.П., Швецов А.Н., Шеер А.В., Уэно Х., Яковлев С.А., Jennings N.R., Minsky M., Wooldridge M.J..

В настоящее время не существует средств динамического моделирования ситуаций (СДМС) мультиагентных процессов преобразования ресурсов, а близкие к данному классу системы, как будет показано в дальнейшем, имеют ряд недостатков: неполный набор функциональных возможностей, отсутствие поддержки функции проектирования концептуальной модели предметной области и построения мультиагентных моделей, содержащих интеллектуальных агентов; неполное соответствие понятийного аппарата проблемной области мультиагентных процессов преобразования ресурсов, отсутствие поддержки русского языка.

В связи с этим актуальным является исследование существующих математических ситуационных моделей процессов, адекватных мультиагентному процессу преобразования ресурсов, и на их основе создание проблемно-ориентированного пакета моделирования, который должен обеспечить возможность ЛПР самостоятельно создавать модели ситуаций и с их помощью решать различные задачи анализа.

Идея книги заключается в интеграции методов и инструментальных средств ситуационного, мультиагентного, имитационного и экспертного моделирования с целью повышения эффективности принятия решений при ситуационном управлении преобразованием ресурсов.

Структура предлагаемого материала выглядит следующим образом.

Книга состоит из введения, четырех глав, заключения, списка литературы и приложений.

В *первой главе* книги обоснована необходимость автоматизации процесса принятия решений (ППР), приведен обзор методов моделирования мультиагентных процессов преобразования ресурсов, рассмотрены системы, близкие по функциональности к СДМС, и выполнен их сравнительный анализ, определены требования к СДМС мультиагентных процессов преобразования ресурсов.

Во *второй главе* проведен анализ существующих ситуационных моделей (семиотической и SIE-модели). Предложена математическая ситуационная модель мультиагентного процесса преобразования ресурсов, созданная на основе интеграции аппаратов процессов преобразования ресурсов, ситуационного и экспертного моделирования, мультиагентных систем.

В *третьей главе* излагаются принципы построения СДМС мультиагентных процессов преобразования ресурсов, приведено описание разработанной мультиагентной СДМС.

В *четвертой главе* рассмотрены примеры применения мультиагентной СДМС процессов преобразования ресурсов при решении следующих задач: моделирование деятельности базового предприятия; анализ и прогнозированиедеятельности производственного предприятия.

Авторы выражают признательность Л.Г. Доросинскому, Б.И. Клебанову, В.М. Кормышеву, В.И. Кузякину, В.И. Суханову, А.Ф. Шорикову, А.Ю. Филипповичу за справедливые замечания и ценные предложения при обсуждении рукописи. За предоставленную экспериментальную базу благодарим С.Ю. Долматова (ЗАО «Уральская индустриальная группа»).

Авторы благодарны Е.Ф. Смолий за оказанную неоценимую помощь при разработке, отладке и внедрении СДМС BPsim2.

1. ОБЗОР МЕТОДОВ И СРЕДСТВ СИТУАЦИОННОГО МОДЕЛИРОВАНИЯ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ

1.1. Ситуационное управление организационно-техническими системами

1.1.1. Ситуационный подход в управлении

В становление теории мультиагентных процессов преобразования ресурсов и развитие систем динамического моделирования существенный вклад внесли работы Борщёва А.В. [1], Вавилова А.А. [2–3], Гольдштейна С.Л. [4–5], Емельянова С.В. [2–3], Исидзуки М. [6], Карпова Ю.Г. [7], Клыкова Ю.И. [8–10], Попова Э.В. [11], Поспелова Д.А. [12–13], Прицкера А. [14], Советова Б.Я. [15], Форрестера Дж. [16–17], Филипповича А.Ю. [18–20], Чистова В.П., Швецова А.Н. [21–23], Шеера А.В. [24–25], Уэно Х. [6], Яковлева С.А. [15, 23], Jennings N.R. [26–28], Minsky M. [29], Wooldridge M.J. [26, 30], и др..

В разделе рассматривается развитие понятия ситуации и ситуационный подход в управлении. Перед тем как рассмотреть понятие ситуация, раскроем базовые понятия моделирования организационно-технических систем (ОТС): система, цель, задача, организационно-техническая система.

Система – совокупность элементов, находящихся во взаимодействии. С понятием «система» неразрывно связаны такие элементы, как цель и задача. Цель системы – достижение и сохранение желаемого состояния или желаемого результата поведения системы. Применительно к организации более подходит следующее определение цели. Цель организации – стремление к максимальному результату, выражаемому в максимизации ценности капитала, при постоянном сохранении определенного уровня ликвидности и достижении целей производства и сбыта с учетом социальных задач. Система целей – совокупность взаимоувязанных целей. Задача системы – описание способа (технологии) достижения цели, содержащего указание на цель с желаемыми конкретными числовыми (в том числе временными) характеристиками [31].

Современная организационно-техническая система является сложной системой управления, включающей многосортные множества взаимосвязанных и взаимодействующих в пространстве и во времени элементов, формирующих её интегративные свойства и функционирующих совместно для достижения целей, поставленных перед системой [21].

Слово «ситуация» используется повседневно в самых различных аспектах и порой неотделимо от таких понятий, как состояние, событие, процесс, положение и т. д. Основоположники ситуационного управления Клыков Ю.И. [8–9] и Поспелов Д.А. [12–13] в своих ранних работах явно отождествляют ситуацию с состоянием. Под ситуацией (дискретной совокупностью) понимается множество транзактов (оперативных элементов), расположенных в определенных точках статической системы [12]. Позднее авторы расширяют понятие, добавляя в него информацию о связях между объектами: *«текущая ситуация – совокупность всех сведений о структуре объекта и его функционировании в данный момент времени»* [13]. Все сведения подразумевают также причинно-следственные связи, которые могут выражаться множеством последовательных событий или процессов. В этом смысле ситуация кардинально отличается от состояния и события, которые могут соответствовать только одному моменту времени [18].

Будем придерживаться следующего определения ситуации, данного в работе [18]: *Ситуация системы есть оценка* (*анализ, обобщение*) *совокупности характеристик объектов и связей между ними, которые состоят из постоянных и причинно-следственных отношений, зависящих от произошедших событий и протекающих процессов.*

Обобщенное описание (отображение) системы с помощью ситуаций называется *ситуационной моделью* (СМ). В связи с этим все ситуационные системы можно называть системами ситуационного моделирования (ССМ). В [18] *под ССМ понимается комплекс программных и аппаратных средств, которые позволяют хранить, отображать, имитировать* (*симулировать*) *или анализировать информацию на основе СМ.*

Принцип *ситуационного управления* базируется на понятии полной ситуации как совокупности, состоящей из состояния (текущей ситуации), знаний о состоянии системы управления в данный момент

и знаний о технологии управления. Элементарный акт управления представлен в следующем виде [18]:

$$S_i : Q_j \xrightarrow[U_k]{} Q_i, \tag{1.1}$$

где S_i – полная ситуация;
Q_i – новая ситуация;
Q_j – текущая ситуация;
U_k – способ воздействия на объект управления (одношаговое решение).

Смысл этого соотношения заключается в следующем [18]: если на объекте управления сложилась ситуация Q_j и состояние системы управления и технологическая схема управления, определяемые S_i, допускают использование воздействия U_k, то оно применяется, и текущая ситуация Q_j превращается в новую ситуацию Q_i.

Данная книга посвящена ситуационному моделированию дискретных процессов преобразования ресурсов.

1.1.2. Рассмотрение организационно-технических систем с точки зрения процессов преобразования ресурсов

В разделе рассматривается предметная область процессов преобразования ресурсов, охватывающая такие классы процессов, как производственные, технологические, организационные, бизнес-процессы и цепочки поставок, и рассматривается возможность применения ситуационного подхода к данной предметной области.

Под процессом преобразования ресурсов понимается непрерывный или дискретный процесс преобразования входа (ресурсов, необходимых для выполнения процесса) в выход (продуктов – результатов выполнения процесса). Элемент (компонент) такого процесса преобразования ресурсов или весь процесс представлен в виде структуры, включающей: вход, условие запуска, преобразование, средства преобразования, выход [33–43].

В процессе преобразования ресурсов обычно происходит уменьшение объема входа и увеличение объема выхода. В момент выполнения условия запуска уменьшается входной ресурс и захватываются средства. В момент окончания преобразования происходит увеличение выходного ресурса и освобождение средств. Таким образом, процесс преобразования ресурсов позволяет описывать большинство окружающих нас процессов. Данный подход лег в основу теории

динамического моделирования процессов преобразования ресурсов, успешно используемой авторами для решения задач управления производственными и бизнес-процессами [33–43].

В качестве примера системы преобразования ресурсов может быть рассмотрено любое производственное предприятие. В обобщенном графическом виде (на верхнем уровне) коммерческое предприятие может быть представлено следующей схемой движения ресурсов предприятия (рис. 1.1) [33]. На этой схеме прямоугольниками обозначены ресурсы предприятия, а овалами преобразователи ресурсов.

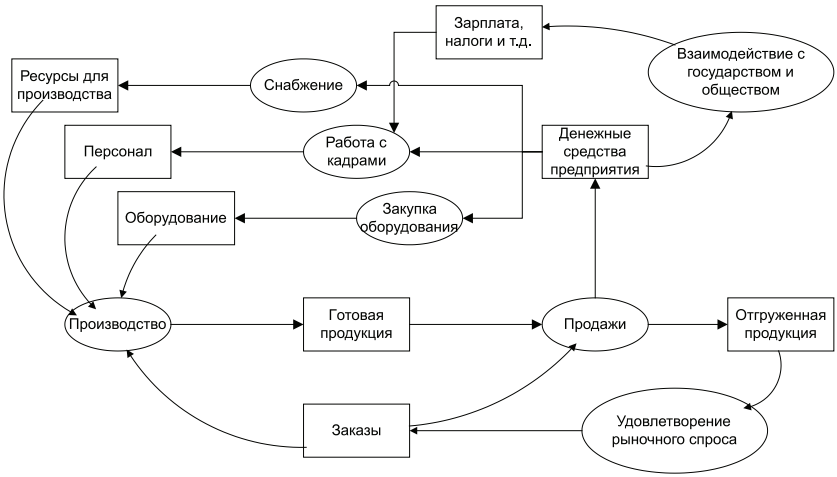


Рис. 1.1. Потоковая модель предприятия

Образовательный процесс вуза в концепции процесса преобразования ресурсов можно представить в следующем виде: вход (абитуриенты / студенты); выход (студенты / специалисты с высшим образованием, знания и навыки); средства (преподаватели, аудитории, лаборатории, литература, спецтехника, и т. д.); условия запуска (планы рабочих групп, учебные планы, расписания занятий) рис. 1.2 [44–47].

В целом можно отметить, что подход к моделированию процессов, используемый авторами теории процессов преобразования ресурсов, близок к подходу рассмотрения процесса в виде полюсника или группы полюсников, используемому в работах следующих авторов Клыкова Ю.И. и Поспелова Д.А. [9–10, 12–13] при моделировании

технологических и производственных процессов, работы морского порта, аэропорта и железнодорожного узла, диспетчеризации тампонажных работ при бурении нефтяных и газовых скважин; Чистова В.П., Кононенко И.А., Ситникова И.О., Захаровой Г.Б. при моделировании и проектировании радиоэлектронной аппаратуры [48–50]. Методы ситуационного управления нашли применение при автоматизации управления процесса капитального строительства [51] на основе моделей сетевого планирования, которые также используются в задачах распределения и планирования ресурсов. С точки зрения ситуационного управления процесс преобразования ресурсов в графическом виде будем представлять следующим образом (рис. 1.3).

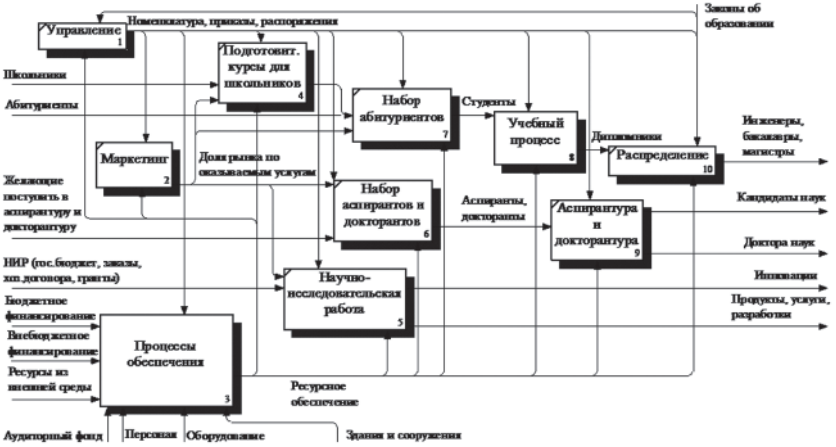


Рис. 1.2. Основные процессы вуза в IDEF0

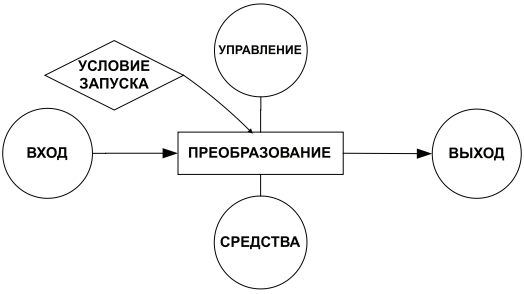


Рис. 1.3. Ситуационное представление процесса преобразования ресурсов

Под элементом «управление» процесса преобразования ресурсов будем понимать набор управляющих воздействий (команд). Условие запуска определяет момент запуска процесса преобразования ресурсов на основании: состояния процесса преобразования, входных и выходных ресурсов, стартующих (запускающих преобразование) команд управления, средств, с помощью которых осуществляется преобразование (далее «средств») и других событий, возникающих во внешней среде процесса. В момент запуска определяется время выполнения преобразования на основании параметров команды управления и имеющихся ресурсных ограничений.

Классификация ресурсов с точки зрения их использования [139] в процессе преобразования приведена на рис. 1.4.

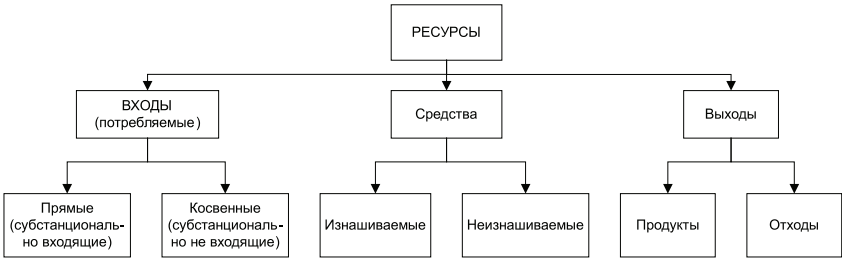


Рис. 1.4. Классификация ресурсов по типам использования

Потребляемые ресурсы (входы) – ресурсы, которые используются в процессе только один раз. В зависимости от роли в процессе преобразования потребляемые ресурсы делятся на прямые (непосредственно входящие в конечный продукт и являющиеся его составной частью) и косвенные (участвующие в процессе преобразования, но не являющиеся составной частью конечного продукта).

Средства не потребляются, а используются в процессе преобразования, они не уменьшаются в процессе их использования (в большинстве случаев используются многократно, в зависимости от их эксплуатационного потенциала). Средства подразделяются на изнашиваемые и неизнашиваемые (снижается потенциал ресурса с течением времени или нет).

Выходы формируются в процессе преобразования. Выходы подразделяются на продукты и отходы.

Объекты ОТС характеризуются сложностью структуры и алгоритмов поведения, многопараметричностью, что, естественно, приводит

к сложности их моделей; это требует при их разработке построения иерархических модульных конструкций, а также использования описания внутрисистемных процессов [15]. Сложные процессы преобразования ресурсов, с точки зрения структурного подхода [2–3, 97–98, 137], могут быть представлены в виде иерархии последовательных декомпозиций (детализаций) процесса на подпроцессы. Каждая декомпозиция представляет собой композицию (состав) более простых элементов процесса преобразования. Тем самым создается иерархическая-многоуровневая модель процесса. На самых нижних уровнях процесс может быть представлен с точностью до элементарных операций преобразования ресурсов.

Согласно принципам создания ситуационных динамических моделей, изложенных Клыковым Ю.И. в [8–9], при построении моделей сложных систем управления целесообразно строить иерархические модели уровней управления, причем на каждом вышележащем уровне формируются команды для нижележащего уровня.

На различных уровнях сложной системы управления используются следующие три основных способа построения композиций команд: последовательный, параллельный и смешанный. При формализации функционирования сложной системы в виде дискретной сети команды управления задаются с помощью графов функционирования автоматов сети, отображающих функциональную и временную структуры команд. Элементарные команды управления характеризуют всевозможные переходы между смежными вершинами графов функционирования автоматов сети. Производные команды управления, являющиеся композициями элементарных, задаются в виде путей графов функционирования автоматов. Введение дискретной сети позволяет дать точное определение команды управления большой системой в момент времени t [8]. *Таким образом, структура композиции команд управления напрямую зависит от структуры процессов системы.*

Наиболее распространенным средством моделирования динамических процессов (переходов из одного состояния в другое (из одной ситуации в другую)) является имитационное моделирование и, в частности, дискретно-событийное [2–3, 8, 14–20, 33–34, 48].

Анализ структуры и функционирования широкого класса сложных систем показывает, что большую систему можно рассматривать как совокупность элементов двух видов: объекты (поезда, самолеты, суда, станки, заводы, железнодорожные узлы, шоссейные дороги,

люди, промышленные сооружения и т. д.) и отношения, характеризующие пространственно-временные связи между объектами. При этом одни элементы сложной системы могут представлять собой совокупность других. Например, завод представляет собой совокупность цехов, расположенных определенным образом в пространстве и связанных между собой технологическими линиями. Каждый цех состоит в свою очередь из участков, специализирующихся по выпуску определенных видов оборудования, и т. д. [8].

Язык, с помощью которого формализуются структура и функционирование сложных систем, должен обладать средствами отображения иерархической структуры большой системы и множества отношений между объектами, а также быть близким к естественному языку, на котором осуществляется содержательное описание управляемого объекта. Простейшими единицами такого языка должны быть модули, из которых строятся все остальные единицы по правилам грамматики языка. Описание связей между объектами, а также законов функционирования отдельных объектов удобно осуществлять с помощью дискретных сетей. Язык описания ситуаций предназначен для формализации состояния дискретной сети, а также точного определения микроситуации управляемого объекта. Благодаря использованию единого языка описания состояний управляемого объекта и его модели обеспечивается возможность имитации структуры объекта и процессов, протекающих на этой структуре [8].

Определяющим моментом построения моделей сложных процессов преобразования ресурсов является, как это было уже ранее отмечено, возможность иерархического представления структуры процесса. Для решения этой задачи в предметной области процессов преобразования ресурсов [33–43] был успешно применен аппарат системных графов высокого уровня интеграции, представленный в работах [2–3].

Специфика больших организационно-технических систем позволяет сформулировать следующие требования к ситуационной модели процесса преобразования ресурсов и средству ситуационного динамического моделирования:

– описание структуры большой системы в виде совокупности элементов и множества отношений между элементами (*семантическая составляющая или возможность построения семантической модели предметной области*)*;*

– представление иерархической структуры процесса (иерархический язык описания предметной области);

– язык описания предметной области и ситуаций, близкий к естественному языку;

– наличие языка описания управляющих воздействий (*команд управления*) *сложной системы.*

1.2. Процесс принятия решений и информационные технологии в организационно-технических системах управления

Принятие решений – каждодневная деятельность человека, часть его повседневной жизни. В большинстве случаев оно заключается в генерации возможных альтернатив решений, их оценке и выборе лучшей альтернативы. При выборе альтернатив приходится учитывать большое число противоречивых требований и, следовательно, оценивать варианты решений по многим критериям. Противоречивость требований, неоднозначность оценки ситуаций, ошибки в выборе приоритетов сильно осложняют принятие решений. Также постоянно меняется круг задач, решаемых человеком в различных сферах своей деятельности. Возникают новые сложные и непривычные для него проблемы. В течение столетий люди могли принимать решения, ориентируясь на один-два главных фактора, не учитывая многие другие. Они жили в мире, где темп изменения окружающей среды был невелик, и новые явления возникали «по очереди», а не сразу. Сейчас большое количество задач, если не большинство, являются **многокритериальными задачами**, в которых приходится учитывать большое число факторов. В этих задачах человеку приходится оценивать множество сил, влияний, интересов и последствий, характеризующих варианты решений [52–53].

В ОТС оказывается довольно сложно оценить параметры потоков информации, установить определенные и нормированные структуры данных для принятия решений. Для систем такого типа характерно вероятностное поведение, вызываемое воздействием множества объективных и субъективных факторов, таких как частые реорганизации и правовая неопределенность; высокая изменчивость источников и адресатов информации, номенклатуры и форм представления документов; слабая формализованность маршрутов и методов обработки информации внутри организации; недостаток квалифицированных специалистов в области информационных технологий [21, 54].

Отсюда вытекает потребность в интеллектуальной системе ППР, которая бы взяла на себя все формализованные функции исполнителей и оказала существенную поддержку при решении трудно формализуемых задач. Организационные задачи во многих случаях не имеют точных алгоритмов решения, а разрешаются в рамках некоторых сценариев, которые в общих чертах хорошо известны исполнителям, но в каждой конкретной ситуации могут частично изменяться. На практике это приводит к хорошо всем знакомым процессам согласований, рассмотрений и т. п. Такие сценарии решения организационных задач весьматрудно описать алгоритмическими моделями; более адекватными оказываются модели представления знаний, позволяющие менять правила поведения и осуществлять логические выводы на основании содержания базы знаний (БЗ). В случае возникновения затруднений при решении задач должны быть проведены переговоры с вышестоящим уровнем, поставившим данную задачу и определившим критерии для её решения [21].

В условиях высокой степени неопределенности внешней и внутренней среды предприятий управление представляет собой адаптационный многошаговый процесс постепенного формирования и осмысления перспективных и текущих целей и возможности их достижения. Сложность проблем управления требует оказания руководителям различных уровней систематической помощи на всех этапах процесса принятия решений на основе современных информационных технологий (ИТ), способных обеспечить возможность определения сравнительной эффективности альтернативных вариантов с учетом широкого диапазона непредвиденных осложнений и изменения экзогенных факторов [55].

Важной составляющей при решении поставленных задач и разбиении их на подзадачи являются знания, наработанные по результатам предыдущей деятельности и результатам анализа накопленной информации. В настоящее время знания законсервированы в головах отдельных людей и совершенно недоступны другим. Приходится затрачивать определенные усилия и время на приобретение знаний, которые уже существуют в готовом виде, поэтому в структуре управления должна быть заложена общая БЗ [21].

На уровне стратегического управления ОТС решаются такие задачи, как анализ и моделирование действий в кризисной ситуации, поиск новых решений стратегических проблем, организация взаимодействия

между информационными системами (базами данных (БД) и БЗ, хранилищами данных (ХД)), справочными и другими системами), подготовка моделей прогноза развития кризисной ситуации и оценка рисков принятия решений, анализ состояния различных областей деятельности предприятия, подготовка вариантов стратегического развития. На уровне тактического управления обеспечивается своевременное доведение информации до экспертов и руководителей, контроль доведения решений до исполнителей и обеспечение контроля исполнения, обеспечение руководителей и экспертов сведениями о состоянии объектов управления и технологических процессов, контроль выполнения планов текущих работ по различным направлениям, подготовка аналитической информации по проблемам, связанным с деятельностью предприятия [21].

Из данного раздела вытекают следующие требования к информационным технологиям ППР:

– отсутствие точных алгоритмов решения организационных задач обосновывает применение БЗ и аппарата экспертных систем (ЭС) для накопления опыта решения задач и использования этих знаний (логического вывода) при диагностировании ситуаций и поиске решений;

– на каждом уровне и ветви модели сложной системы управления может находиться модель лица, принимающего решения (ЛПР);

– доступ к понятиям предметной области (к уровню детализации их свойств) необходимо обеспечивать с учетом уровня знаний и компетентности ЛПР, т. е. привязки к уровню и ветви организационного управления;

– требования к модели ЛПР: наличие знаний; сценарии поведения; постановка целей; участие в обмене информацией (сообщениями); механизм управления некоторым множеством элементов модели;

– наличие механизма маршрутизации сообщений между ЛПР.

1.2.1. Системы поддержки принятия решений (СППР)

Проследим развитие определения СППР, которые давали различные авторы. Так, в 80-е годы прошлого века под банком данных для принятия решений понимался *адаптивный человеко-машинный банк с переменной структурой информации, операционная система которого содержит средства определения целевой (прагматической) направленности данных, их анализа, обобщения, прогнозирования и использования под заданные цели и задачи потребителей, а также средства, с помощью*

которых строятся целесообразные семиотические модели мира и формируется наилучшее поведение в мире с помощью этих моделей [10]. Данное определение является наиболее ранним из всех и определяет направленность СППР на решение определенных задач (проблемную ориентацию).

В работах [56–57] приводится следующее определение: СППР – это человеко-машинная информационная система, используемая для поддержки действий ЛПР в ситуациях выбора, когда невозможно или нежелательно иметь автоматическую систему представления и реализации всего процесса оценки и выбора альтернатив. Во-первых, такие системы выступают в роли помощника ЛПР, который позволяет расширить его способности, но не заменяет его мнение и систему предпочтений. Во-вторых, они предназначены для использования в ситуациях, когда ППР ввиду необходимости учета субъективного мнения ЛПР не может быть полностью формализован и реализован на ЭВМ.

Увеличение объема информации, которую необходимо обрабатывать ЛПР, возрастание сложности решаемых задач в условиях необходимости учета большого числа взаимосвязанных факторов и высокой динамики внешней среды, привели к увеличению требований к классу СППР [21, 52–53]. Различные определения данного термина, приведенные в работах [58–60], обобщил Э.А. Трахтенгерц [52]: СППР – это система, выполняющая следующие функции [52]:

1) оценка обстановки (ситуаций), выбор критериев и оценка их относительной важности;

2) генерация возможных решений или сценариев действий;

3) оценка сценариев, решений, действий и выбор наилучших из них;

4) обеспечение информационного обмена и согласование групповых решений;

5) моделирование принимаемых решений при наличии такой возможности;

6) динамический анализ возможных последствий принимаемых решений;

7) сбор данных о результатах реализации принятых решений и оценка этих результатов.

Генерацию возможных решений (сценариев) можно осуществить посредством [52] программной реализации аналитических или имитационных моделей; с использованием экспертных систем; генерации сценариев путем комбинации различных операций, заданных

руководителем или взятых из базы данных; и, наконец, используя подход, получивший название ситуационного управления.

Как следует из раздела, в СППР могут использоваться средства имитационного и ситуационного моделирования, экспертных систем.

1.2.2. СППР в стратегическом управлении

При стратегическом управлении необходимость обрабатывать огромное количество внешней и внутренней информации требует разработки и внедрения в управление предприятием ИТ, позволяющей руководителю [45]:

– получать непрерывную, объективную картину состояния предприятия в целом и его структурных подразделений;

– выявлять тенденции развития предприятия, т. е. понять, к чему оно придет в будущем, если не произойдет каких-либо кардинальных изменений;

– получать ответы на вопросы «что будет, если» и «что надо, чтобы»;

– проводить оценку рисков;

– отслеживать изменения, происходящие с внешней средой и ее влияние на внутренние процессы предприятия;

– планировать и проводить текущие производственные совещания на расширенном информационно-аналитическом базисе.

Степень успешности реализации стратегического планирования и управления зависит от умения адекватно определять соответствие достигнутых результатов поставленным целям, а это, в свою очередь, обуславливается выбранной технологией оценки разнообразных параметров деятельности организации. Таким инструментом является методика Balanced Score Card (BSC, система сбалансированных показателей) – инструмент управления стратегией организации, рассматривающий его деятельность не только по финансовым показателям, но и по качеству работы с клиентами, персоналом, информационными технологиями, производственными процессами и др. Возможные стратегические цели и критерии оценки деятельности организации объединяются в 4 группы: финансы; клиент; внутренние бизнес-процессы (БП); развитие и обучение персонала. Зависимость одних целей от других задается причинно-следственными связями.

BSC проецируется на всю организацию с последующей декомпозицией стратегических целей до структурных подразделений путем разработки индивидуальных задач в рамках уже разработанных

корпоративных стратегий. BSC стимулирует понимание сотрудниками своего места в стратегии развития организации. Основное назначение BSC – обеспечение сбора, систематизации и анализа информации, необходимой для принятия стратегических управленческих решений. Опыт применения BSC в системах поддержки принятия решений и непрерывных имитационных моделях БП описан в [61].

BSC представляет совокупность миссии, видения, мероприятий, стратегий, ключевых показателей деятельности и в целом описывается следующей кортежной моделью:

$$BSC = \langle \text{Миссия, Виденье, Мероприятия, Стратегии, Цели, Показатели, Связи} \rangle. \quad (1.2)$$

Методика системы сбалансированных показателей не позволяет количественно или качественно описать причинно-следственные связи и применить какой-либо математический аппарат для решения задач стратегического управления, таким образом, в данной работе решается задача обеспечения стратегической методики соответствующим математическим аппаратом и проблемно-ориентированным программным обеспечением. Решение данной задачи позволит в процессе генерации решений с использованием СППР анализировать и оценивать эффективность решений в проекции BSC.

BSC предоставляет ОТС лишь формат для выражения ее миссии и стратегии по отдельным направлениям деятельности через систему конкретных целей и показателей. Однако сама по себе она не обеспечивает сбор и обработку необходимой информации и доведение ее до персонала и контрагентов. Устранение выше описанных недостатков предполагается достичь на основе информационных технологий.

Одним из перспективных направлений развития СППР является усиление проблемной ориентации в предметной области стратегического управления за счет реализации средств поддержки методики стратегического управления, например системы сбалансированных показателей.

1.3. Системы ситуационного моделирования

По назначению системы ситуационного моделирования (ССМ) можно разделить на три основных класса [18, 62–66]: *системы ситуационного отображения информации (ССОИ), системы динамического моделирования ситуаций (СДМС) и аналитические ситуационные системы (АСС).*

ССОИ можно разделить на два подкласса: *ситуационные центры* (*СЦ*) *наблюдения* (*отображения – СЦО*), *ССОИ с удаленным доступом* (*распределенные – РССОИ*). Типичными для ССОИ являются задачи наблюдения на достаточно большом ареале земной поверхности; управления (навигации) динамическими объектами; наблюдения (управления) за сложными технологическими процессами (например, на атомных электростанциях); управления сложными транспортными узлами.

Основная задача СЦО – строить изображения ситуаций, возникающих в предметной области, на основе которых оперативный состав принимает управляющие решения в рамках определенных задач.

Специально разработанных СДМС в настоящее время практически не существует, поэтому вместо них адаптируют и используют другие классы систем. В связи с этим СДМС можно разделить на два класса: *специализированные и адаптированные*. Для динамического моделирования (имитации) ситуаций можно использовать два подхода: первый – задание исходных данных и последующий анализ возникающих ситуаций в ССОИ или АСС; второй – представление ситуаций, их взаимосвязей и очередности возникновения с помощью систем имитационного (динамического) моделирования. К адаптированным системам, реализующим первый подход, можно отнести *экспертные системы мониторинга* (*ЭС реального времени*). Примером может служить ЭС G2 (Gensym), которая в своем составе имеет модуль генерации исходных данных. Для воплощения второго подхода лучше всего использовать событийно-ориентированные или комбинированные системы имитационного моделирования [18].

К АСС относятся [18] *системы ситуационного управления* (*ССУ*), некоторые *аналитические ситуационные центры* (*АСЦ*) *и* ЭС *мониторинга*. ССУ реализуют принцип ситуационного управления, описанный выше. Основное отличие ситуационного центра от традиционных систем автоматизации управления состоит в том, что в процессе проведения производственно-управленческого совещания в режиме реального времени можно просчитывать и анализировать последствия управленческих решений [47].

Примерами ситуационных центров являются:

– IBS Центр принятия решений – СЦ Министерства природных ресурсов РФ [67], построенный на базе mySAP ERP;

– мобильный пункт управления для МЧС России [68];

– ситуационно-кризисный центр Минатома [69];

– СЦ производственного предприятия «Кант» [70];
– «Триумф-Аналитика» фирмы Парус [71];
– СЦ региональной энергетической комиссии [72].

Данная работа посвящена разработке моделей, методов и программных средств принятия решений, основанных на методах ситуационного (семиотического) управления. Классификация ситуационных систем, предложенная А.Ю. Филипповичем в [85] на рис. 1.5 была расширена блоками, выделенными серым цветом. Данная классификация показывает разнообразие ситуационных систем. Однако, как показало исследование, **специально разработанных систем динамического моделирования ситуаций (СДМС) в настоящее время практически не существует**, поэтому вместо них адаптируют и используют другие классы систем. СДМС разделяют на два класса: *специализированные* (ориентированные на решение определенной задачи конкретной предметной области) и *адаптированные* (построенные на основе или системы имитационного моделирования (СИМ), или на основе динамической экспертной системы (ЭС)).

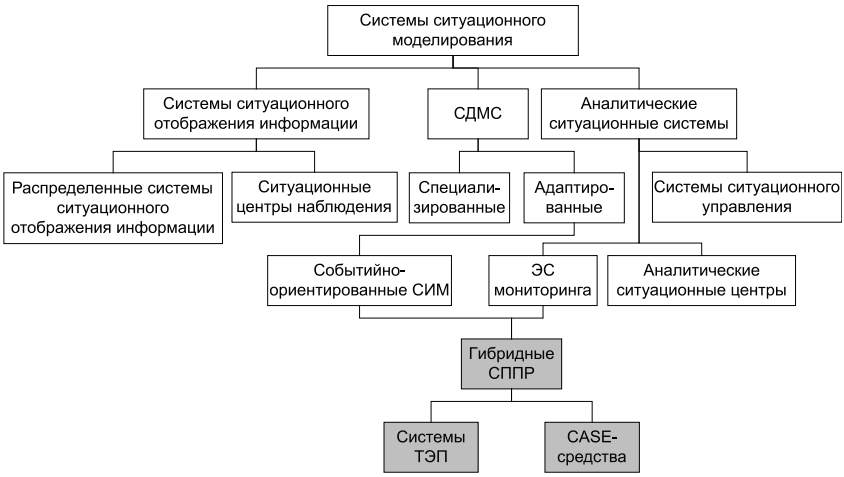


Рис. 1.5. Классификация ситуационных систем

К новым блокам относятся гибридные СППР, построенные в результате интеграции различных методов (в данной работе решается задача интеграции методов имитационного, экспертного, ситуационного

и мультиагентного моделирования, а также объектно-ориентированного подхода). Гибридные СППР ориентированы на решение трудноформализуемых задач принятия решений и управления ОТС, к которым относятся: технико-экономического проектирование (ТЭП) и обоснование, планирование, реинжиниринг, анализ «Что, если». Системы ТЭП ОТС, а также CASE-средства (Computer Aided Software Engineering) [151–155] применяют в своей основе аппараты ЭС и имитационного моделирования (ИМ), адаптированные для решения практических задач.

В перспективе класс РССОИ может пополниться развитыми ERP-системами [62–63] и АСУ ТП. Примером такой системы, реализованной при участии автора является автоматизированная система выпуска металлургической продукции (АС ВМП) [156–181], которая так же включает себя модули, относящиеся к классу СДМС (модуль создания моделей процессов предприятия и оптимизации процессов предприятия [156–179]), к классу АСС относится модуль подготовки данных [180–181], к классу ССУ относится модуль интеграции моделей [179]. Поэтому АС ВМП может быть рассмотрена как пример реализации системы ситуационного моделирования.

1.4. Методы моделирования

В разделе приводится обзор существующих методов в области имитационного, ситуационного и экспертного моделирования.

1.4.1. Имитационное моделирование

В качестве основы для создания СДМС в данной книге предлагается использовать средства и методы имитационного моделирования (ИМ).

Слишком часто в нашей жизни эксперименты ставятся на реальных системах, будь то экономика страны, отдельная организация или система управления сложным перекрестком. Лицо, принимающее решение, надеется при этом на свою проницательность, интуицию и удачу. Еще в 1969 г. известный ученый, родоначальник системной динамики Джей Форрестер отмечал, что на основе интуиции для управления сложными системами чаще выбираются неверные решения, чем верные [16–17], и это происходит потому, что в сложной системе причинно-следственные отношения ее параметров не являются простыми и ясными. Невысокая эффективность управленческих решений, сделанных на основе интуиции, объясняется тем, что

причины и следствия в сложных системах разнесены во времени и пространстве, поэтому человеку трудно предсказать, какие последствия вызовет то или иное решение. В тех случаях, когда для оценки принимаемых решений эксперимент с реальными системами невозможен либо слишком дорог, используется имитационное моделирование [7].

Имитационное моделирование может применяться в самых различных сферах деятельности. Ниже приведен список задач, при решении которых моделирование особенно эффективно [73]:

– проектирование и анализ производственных систем;

– оценка различных систем вооружений и требований к их материально-техническому обеспечению;

– определение требований к оборудованию и протоколам сетей связи;

– определение требований к оборудованию и программному обеспечению различных компьютерных систем;

– проектирование и анализ работы транспортных систем, например аэропортов, автомагистралей, портов и метрополитена;

– оценка проектов создания различных организаций массового обслуживания, например центров обработки заказов, заведений быстрого питания, больниц, отделений связи;

– модернизация различных процессов в деловой сфере;

– определение политики в системах управления запасами;

– анализ финансовых и экономических систем.

Моделирование насчитывает в настоящее время четыре основных направления: моделирование *динамических систем* [14–15]*, дискретно-событийное* моделирование [2–3, 14–15, 73], *системная динамика* [16–17] *и агентное* моделирование [1, 7, 21–22]. В каждом из этих направлений развиваются свои инструментальные средства, упрощающие разработку моделей и их анализ. Данные направления (кроме агентного моделирования) базируются на концепциях и парадигмах, которые появились и были зафиксированы в инструментальных пакетах моделирования несколько десятилетий назад и с тех пор не менялись [7].

Карпов Ю.Г. в [7] отмечает, что проблемы анализа современных реальных систем часто требуют разработки моделей, не укладывающихся в рамки «одной единственной парадигмы моделирования». Новые требования, выводящие за рамки традиционных парадигм при построении моделей, требуют использования скриптовых языков, тонких и сложных средств интеграции внешних программных

модулей с моделью и т. п., что существенно усложняет разработку моделей в традиционных средах [7].

Имитационная модель организационно-технической системы в силу сложной структуры должна быть иерархической, что позволит в свою очередь применять к ней теории иерархических и мультиагентных систем.

Теоретической базой создания средств ИМ являются широко распространенные математические схемы описания динамических процессов (расширенные сети Петри [2, 15], системы массового обслуживания [15, 74], модели системной динамики [16–17]). Новый подход к моделированию динамических процессов, к которым относятся цепочки поставок (логистика [75]), технологические, производственные, организационные и бизнес-процессы, предлагает концепция процессов преобразования ресурсов [33–43], синтезированная на базе вышеупомянутых математических схем.

Системы имитационного моделирования (СИМ) можно разделить на два класса – универсальные и проблемно-ориентированные. Проблемно-ориентированные СИМ имеют одно важное преимущество – они снижают требования к конечному пользователю в области программирования, т. е., с точки зрения внедрения и применения на предприятиях, в организациях и бизнесе имеют больший шанс на выживание. Результаты разработки СИМ, ориентированной на моделирование ситуаций в производственно-экономических системах, представлены в [76]. К распространенным в настоящее время проблемно-ориентированным СИМ в области дискретных процессов преобразования ресурсов относятся следующие: AnyLogic [77], Arena [78–79], ARIS [5, 24–25, 80], ReThink [81–82], Simio, Plant Simulation.

1.4.2. Экспертное моделирование

В работе [18] выделен класс систем экспертного моделирования (СЭМ), которые имитируют процессы рассуждения человека. Исходными данными для них являются декларативные и процедурные знания, поэтому их также называют *системами, основанными на знаниях* (*knowledge-basedsystem*), или *экспертными системами* (*ЭС*). В общем случае ЭС нельзя рассматривать как СИМ, так как они используют критерии, стратегии выбора правил, формализованные цели и т. д. Тем не менее, при моделировании знаний эксперта, которые представляют собой вербальное или графическое отображение системы,

ее связей и закономерностей, экспертное представление аналогично имитационному. Тенденция сближения классов ЭС и СИМ отмечена в работах [18, 83–85].

ЭС – наиболее распространенный класс ИТ, ориентированный на тиражирование опыта высококвалифицированных специалистов в областях, где качество принятия решений традиционно зависит от уровня экспертизы, например, медицина, юриспруденция, геология, экономика, военное дело, энергетика, металлургия, логистика, проектирование. ЭС эффективны лишь в специфических «экспертных» областях, где важен эмпирический опыт специалистов [86].

Структура ЭС может включать следующие компоненты: *база знаний; база данных; машина вывода; интерфейс с пользователем; модуль извлечения знаний и обучения; компонент приобретения и объяснения знаний* [11, 18], первые три – являются обязательными.

База данных (БД) хранит исходные и промежуточные данные решаемой в текущий момент задачи.

База знаний (БЗ) предназначена для хранения долгосрочных данных, описывающих рассматриваемую область, и правил, описывающих целесообразные преобразования данных в этой области.

Отличие БЗ от БД определяют исходя из типа хранимых знаний: в БЗ записывают правила (процедурные знания), а в БД – данные (декларативные знания). Все знания стремятся хранить единообразно, используя один язык представления знаний (ЯПЗ).

Машина вывода, используя исходные данные и знания, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

Интерфейс с пользователем ориентирован на организацию общения со всеми категориями пользователей как в ходе решения задач, так и в ходе приобретения знаний, объяснения результатов работы.

Модуль извлечения знаний и обучения автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом, а также формирует знания на основе анализа прикладных ситуаций.

Компонент приобретения и объяснения знаний объясняет, как система получила решение задачи (или почему она не получила решения) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Моделирование в ЭС (СЭМ) представляет собой *вывод на знаниях*. Механизм вывода во многом зависит от используемого ЯПЗ и может быть логическим, нечетким, вероятностным, продукционным и т. д.

Методы имитации символической модели ЭС практически полностью совпадают с методами, используемыми в СИМ. Такое сближение имитационного и экспертного подхода приводит к идеям интеграции ЭС и СИМ, более подробно обсуждаемой во второй главе и рассматриваемых работах [18, 33, 87–88].

Особо следует выделить методы комбинированного вывода, которые учитывают возможности различных ЯПЗ. Кроме того, в ЭС часто используются методы с вызовом внешних процедур (программ) и получения из них данных. Эти методы носят название *процедур извлечения знаний* [18].

1.4.3. Ситуационное моделирование

Одним из перспективных направлений создания моделей принятия решений, позволяющим использовать содержательные сведения о конкретных ситуациях и отражать реальную динамику процессов, а также учитывать человеческий фактор в процессе выбора решений, является метод ситуационного управления. Метод оформился в начале 70-х годов трудами российских ученых В.Н. Пушкина, Д.А. Поспелова, Ю.И. Клыкова, Э.Ф. Скороходько как реакция на трудности применения точных количественных методов, в частности математического программирования народнохозяйственными объектами [5]. Для описания ситуаций используются *семиотические* (*ситуационные*) *языки и модели,* среди которых можно выделить следующие основные подходы [18]:

– *дискретные ситуационные сети* (*ДСС*);
– RX-коды;
– *логика предикатов;*
– *универсальный семантический код.*

ДСС представляет собой сложную семантическую сеть [10]. Каждая ситуация описывается ориентированным графом (сетью), а для представления вложенности («ситуации ситуаций») используются гиперграфы, т. е. некоторый фрагмент семантический сети, определяющий ситуацию, который может рассматриваться как одна вершина сети.

RX-коды представляют собой язык бинарных отношений и имеют в качестве ядерной конструкции запись следующего вида [89]:

$$x_1 = x_2 r_2 x_3 r_3, \tag{1.3}$$

где x_i – объект или ситуация;

r_i – отношение.

Логика предикатов – раздел математической логики, изучающий логические законы, общие для любой области объектов исследования (содержащей хоть один объект) с заданными на этих объектах предикатами (т. е. свойствами и отношениями).

Универсальный семантический код использует в качестве ядерной конструкции тройку *SAO,* которая соответствует *субъекту S,* совершающему *действие A над объектом O.*

Для реализации в ЭВМ семиотических языков используют языки представления знаний. Наиболее близким подходом к описанию семиотических конструкций является семантическая сеть. Однако сети очень медлительны при использовании операций поиска, поэтому конструкции часто представляют с помощью логики предикатов [11], фреймов [6, 11, 29,90] и продукций [86, 91].

В [18] отмечается, что методы представления знаний в ситуационных системах и ЭС аналогичны. Еще больше они сблизились после активного внедрения нечеткой логики в технологии ЭС.

При ситуационном моделировании активно используются имитационные модели, следовательно, ситуационный *«язык должен включать некоторые средства, присущие языкам моделирования: системное время, очереди событий, организацию квазипараллельных процессов и т. д.»* [10].

Интерпретируя определение ситуации Филипповича А. Ю. применительно к процессам преобразования ресурсов, под ситуацией будем понимать оценку совокупности характеристик объектов (образуемых на множестве элементов процесса преобразования ресурсов, средств, операций, процессов, команд управления и т. д.) и связей между ними, которые состоят из постоянных и причинно-следственных отношений, зависящих от прошедших событий и протекающих процессов.

1.4.4. Мультиагентный подход

Для решения задачи построения моделей ЛПР на разных уровнях сложной системы (см. раздел 1.2), целесообразно использовать

теорию мультиагентных систем, новое направление развития искусственного интеллекта, информационно-телекоммуникационных технологий и имитационного моделирования. Ниже приводится краткий обзор результатов данного направления.

Агентно-ориентированный подход уже нашел применение в таких областях, как распределенное решение сложных задач, реинжиниринг предприятий, телекоммуникации, электронный бизнес [92], проектирование [93] и т. п. Важной областью применения мультиагентных технологий является моделирование. В этой области Д.А. Поспелов [94] выделяет два класса задач:

1) задачи распределенного управления и задачи планирования достижения целей, где усилия разных агентов направлены на решение общей проблемы и необходимое обеспечение эффективного способа кооперации их деятельности;

2) задачи, где агенты самостоятельно решают свои локальные задачи, используя общие, как правило, ограниченные ресурсы [92].

Термин «агент» происходит от латинского глагола agere, что означает «действовать», «двигать», «править», «управлять» [95]. Понятие *агент* соответствует аппаратно или программно реализованной сущности, которая способна действовать в интересах достижения целей, поставленных перед ней владельцем и/или пользователем, и которая обладает определенными интеллектуальными способностями [21, 92]. В дальнейшем будем придерживаться данного определения.

Две базовые характеристики – автономность и целенаправленность – позволяют отличать интеллектуального агента (ИА) от других программных и аппаратных объектов (модулей, подпрограмм, процедур и т. п.).

Наиболее известными исследовательскими центрами в области агентных систем и технологий являются университет Карнеги Мэллон (Carnegi Mallon University), Массачусетский университет (University of Massachusetts at Amherst), университет г. Болоньи (Univrsita di Bologna), ряд университетов и колледжей Великобритании (Stanford University, Manchester Metropolitan University). Занимаются этими проблемами и крупные корпорации (IBM, Microsoft, DEC, Apple, Toshiba, HewlettPackard и др.). В нашей стране исследования по данной тематике проводятся в Исследовательском центре искусственного интеллекта Института программных систем РАН (г. Переславль-Залесский), в Институте проблем управления РАН,

в Санкт-Петербургском институте информатики и автоматизации РАН, в Санкт-Петербургском государственном электротехническом университете, в Санкт-Петербургском техническом университете, в Институте проблем управления сложными системами РАН (г. Самара), в Уфимском государственном авиационно-техническом университете, в Таганрогском радиотехническом университете [21].

Интеллектуальная мультиагентная система представляет собой множество интеллектуальных агентов, распределенных в сети, которые мигрируют по ней в поисках релевантных данных, знаний, процедур и кооперируются для достижения поставленных перед ними целей [92].

Основными направлениями научного поиска являются [21] теории агентов, которые рассматривают математические методы и формализмы абстрактного представления структуры и свойств агентов и способы построения рассуждений в таких системах; методы коллективного поведения агентов; архитектуры агентов и МАС; методы, языки и средства коммуникации агентов; языки программирования агентов; методы и средства автоматизированного проектирования МАС; методы и средства обеспечения мобильности агентов. Известные подходы проектирования агентно-ориентированных систем можно разделить на две группы [21]:

– базирующиеся на объектно-ориентированных методах и технологиях с использованием соответствующих расширений;

– использующие традиционные методы инженерии знаний.

В методологиях первой группы разрабатываются расширения объектно-ориентированных методов и технологий для проектирования агентно-ориентированных систем. Существует ряд CASE-средств [96–104], поддерживающих объектно-ориентированные методы разработки ИС, среди которых наиболее известными являются AllFusion [97–99, 101–102] фирмы Computer Associate и Rational Rose фирмы IBM [5, 99–100, 102–104], процесс проектирования в которых основывается на языке объектно-ориентированного проектирования UML [100, 102–105]. Современное средство ИМ AnyLogic, поддерживающее агентный подход, использует расширение языка UML-RT [1, 7, 77, 106–108]; подробнее данный инструментарий рассматривается в разделе 1.5.

Вторая группа методологий строится на расширении традиционных методов инженерии знаний [21]. Эти методологии обеспечивают формальные и композиционные языки моделирования для

верификации структуры системы и функций. Эти подходы хорошо применимы к моделированию знаний и информационно-ориентированных агентов.

Заканчивая обзор методов моделирования, необходимо отметить то, что специально разработанных СДМС в предметной области процессов преобразования ресурсов не существует, поэтому актуальным является разработка СДМС, поддерживающая следующие функциональные возможности:

– описание ситуационной модели в виде дискретной ситуационной сети, как наиболее соответствующей процессам преобразования ресурсов;

– декомпозицию ситуационной модели;

– представление информации и моделей с использованием когнитивной графики;

– описание моделей ЛПР в виде интеллектуальных агентов, обрабатывающих информацию, диагностирующих ситуации, вырабатывающих решения (работающих со знаниями), действующих в соответствии с найденным решением и своей моделью поведения (обладающих моделью поведения), участвующих в обмене сообщениями с другими агентами;

– вывод на знаниях;

– имитационное дискретно-событийное моделирование.

В рамках данной работы ставится и решается задача интеграции следующих математических аппаратов: имитационного моделирования (дискретных процессов преобразования ресурсов), *экспертных систем, ситуационного* и мультиагентного моделирования.

1.5. Обзор и сравнение систем динамического моделирования ситуаций (СДМС)

1.5.1. Обзор существующих СДМС

В настоящее время не существует систем подобного класса, вместо них адаптируют другиеклассы систем, частично реализующие требуемые функции и математические аппараты. Ниже приводится обзор близких по функциональности систем.

ARISToolSet – система проектирования и моделирования бизнес-процессов. Обзор сделан по материалам [5, 24–25, 80]. Система ARIS представляет собой интегрированную среду анализа и моделирования. ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы: организационные; функциональные; информационные; модели управления.

В рамках каждого из перечисленных типов создаются модели разных видов, отражающие соответствующие стороны исследуемой системы. ARIS поддерживает большое количество методов моделирования, используемых для построения этих моделей. Среди них такие известные, как диаграммы Чена, Unified Modeling Language (UML), Object Modeling Technique. Для описания БП в ARIS используется общая ARIS-модель процесса [24–25]. В дополнении к общей ARIS-модели для описания процессов используется стандарт EPC (extended Event Driven Process Chain) – расширенная нотация описания цепочки процесса, управляемого событиями.

Экспертная система реального времени G2. Фирма Gensym предлагает графическую, объектно-ориентированную среду для создания интеллектуальных прикладных программ, которые контролируют, диагностируют и управляют динамическими событиями в сетевых и моделируемых средах. G2 для создания правил, моделей и процедур использует структурированный естественный язык. Экспертная система G2 является основой всех прикладных программ фирмы Gensym. *Пакет прикладных программ (ППП) ReThink* предназначен для разработки приложений в области организационного управления и обеспечен графической средой проектирования моделей, объектно-ориентированной подсистемой имитации для тестирования этих моделей и инструментарием для измерения временных, стоимостных и других показателей эффективности производства. ППП ReThink является проблемно-ориентированным приложением комплекса G2, которое позволяет разработчикам использовать не только специализированные средства моделирования процессов, но и универсальные средства комплекса по созданию интеллектуальных объектно-ориентированных систем реального времени [11, 81–82, 109].

Система имитационного моделирования AnyLogic

Система AnyLogic – одна из немногих российских разработок в области ИМ, получивших признание за рубежом. AnyLogic представляет собой среду для графического создания моделей с использованием объектно-ориентированного языка Java. После создания модели и описания экспериментов автоматически генерируется программа имитационного моделирования [1, 7, 77, 106–107, 110–117].

AnyLogic поддерживает на единой платформе следующие существующие подходы: дискретно-событийного и непрерывного

моделирования (блок-схемы процессов, системную динамику, агентное моделирование, карты состояний, системы уравнений). Описание поведения объектов производится с помощью фрагментов кода на языке Java. При разработке сложных моделей не удается обойтись без процедурной логики и, как следствие, написания значительного объема программного кода. Доля программирования в этом случае составляет примерно 80 % общих трудозатрат на разработку модели.

Система имитационного моделирования BPsim

BPsim [33–37, 41–43] – проблемно-ориентированная система ИМ, позволяющая адекватно описывать и моделировать экономические, производственные, технические, информационные и бизнес-процессы в рамках процессов преобразования ресурсов. СИМ BPsim обеспечивает выполнение следующих функций [33–37, 39, 41–47]:

a) создание динамической модели ППР;

b) имитационное моделирование;

c) анализ результатов имитационного эксперимента;

d) получение отчетов по моделям и результатам экспериментов;

e) экспорт результатов экспериментов в MS Excel и MS Project.

Встроенный математический аппарат используется для описания процессов преобразования ресурсов: функций условия запуска, входа, выхода, обработки сообщений. Синтаксически правила задаются с помощью графико-синтаксических диаграмм (интерфейсов), ориентированных на проблемную область процессов преобразования. В качестве операндов используются множества: ресурсов, средств, заявок, сообщений, параметров.

Для описания структуры процессов используются операторы:

a) декомпозиции (детализирует сложный процесс на композицию более простых);

b) сопряжения элементов подмодели (в графическом виде задает причинно-следственные связи между элементами подмодели).

1.5.2. Требования к СДМС

На основе предыдущих разделов выделим следующие требования к средствам СДМС:

1. Проектирование концептуальной модели предметной области.
2. Описание динамических процессовпреобразования ресурсов:
   a) Описание ресурсов, средств, преобразователей.

b) Описание целей (в виде графа, в виде карты BSC).

c) Поддержка создания иерархической модели процесса.

3. Наличие языка описания команд.

4. Возможность описания модели на ограниченном естественном языке.

5. Возможность построения мультагентных моделей. Наличие агентов (моделей ЛПР), обладающих моделью поведения и знаниями:

a) Наличие класса (элемента) «агент», на основе которого можно создавать ИА.

b) Модель поведения агента (язык описания сценариев поведения агентов, язык описания команд управления).

c) База знаний агента.

d) Язык обмена сообщениями между агентами.

6. Поддержка ИМ илиинтегрируемость СДМС с СИМ.

7. Поддержка экспертного моделирования (ЭС). Описание знаний о предметной области. Данная функция необходима для накопления знаний и последующего вывода на знаниях.

8. Поддержка ситуационного подхода. Наличие языка описания ситуаций.

1.5.3. Сравнительный анализ СДМС

В табл. 1.1 приводятся результаты сравнительного анализа.

Как следует из табл. 1.1 и проведенного сравнительного анализа (приложение) ни одна из рассмотренных систем не обладает полной функциональностью мультиагентной СДМС процессов преобразования ресурсов. На основе систем G2 и AnyLogic возможно построение мультиагентных СДМС, причем значительно меньшие усилия потребуются при использовании G2, так как данная система поддерживает аппарат ЭС. Функции проектирования концептуальной модели предметной области и построения мультиагентных моделей, содержащих интеллектуальных агентов, рассмотренные системы не поддерживают. Средствами поддержки методики BSC обладает только система ARIS, но она не поддерживает интеграцию ИМ и BSC. Понятийный аппарат всех систем соответствует проблемной области процессов преобразования ресурсов. Описание модели на ограниченном естественном языке (английском) поддерживается в системе G2. С точки зрения пользователя, не обладающего навыками программирования, удобными средствами описания/создания модели

мультиагентного процесса преобразования не обладает ни одна из систем. В системах AnyLogic и G2 при создании сложных мультиагентных моделей графических средств недостаточно, приходится использовать программный код. К достоинствам пакетов AnyLogic и G2 можно отнести использование языков высокого уровня (ЯВУ), благодаря чему пакеты могут предоставлять разработчику моделей серьезный уровень функциональности.

Таблица 1.1

Сравнительный анализ систем близких по функциональности к СДМС

№ п/п	Параметр	ARIS	G2	AnyLogic	BPsim
1	Проектирование концептуальной модели предметной области	НЕТ	НЕТ	НЕТ	НЕТ
2	Язык описания процессов преобразования ресурсов				
2.1	Описание ресурсов, средств, преобразователей	+	+	+	+
2.2	Описание целей системы – в виде графа; – в виде BSC.	 + +	 + НЕТ	 НЕТ НЕТ	 НЕТ НЕТ
2.3	– Иерархическая модель процесса	+	+	+	+
3	Наличие языка описания команд	НЕТ	+	НЕТ	НЕТ
4	Описание модели на ограниченном естественном языке	НЕТ	+	НЕТ	+
5	Построение мультиагентной модели				
5.1	– Элемент АГЕНТ	НЕТ	НЕТ	+	НЕТ
5.2	– Модели поведения агентов	НЕТ	НЕТ	+	НЕТ
5.3	– База знаний агента	НЕТ	НЕТ	НЕТ	НЕТ
5.4	– Язык обмена сообщениями	НЕТ	НЕТ	НЕТ	НЕТ
6	Имитационное моделирование	+	+	+	+
7	Экспертное моделирование	НЕТ	+	НЕТ	НЕТ
8	Ситуационный подход	НЕТ	+	НЕТ	НЕТ

С целью уменьшения времени, необходимого на интерпретацию результатов эксперимента ЛПР, целесообразно дополнить СДМС автоматизированными средствами поддержки методики BSC. Как следствие, одним из требований к проблемно-ориентированным СДМС, используемым в области стратегического управления, является наличие интерфейса стыковки с типовым ПО, поддерживающим методику BSC, или внутреннего специализированного инструментария.

Выводы

Проблема исследования и разработки моделей, методов и средств динамического моделирования мультиагентных систем в настоящее время находится в стадии развития. На данный момент не существует СДМС для предметной области процессов преобразования ресурсов. Об актуальности исследования, разработки и внедрения методов, моделей и средств динамического моделирования ситуаций говорится в работах Андрейчикова А.В., Андрейчиковой О.Н., Клыкова Ю.И., Поспелова Д.А., Трахтенгерца Э.А., Филипповича А.Ю., Швецова А.Н.

Требуют дальнейшего исследования и разработки такие вопросы, как:

1. Создание динамической ситуационной модели мультиагентного процесса преобразования ресурсов на основе интеграции аппаратов ИМ, ЭС, ситуационного управления и мультиагентных систем.

2. Реализация языковых и инструментальных средств (проблемно-ориентированных) динамического моделирования ситуаций в области мультиагентных процессов преобразования ресурсов, поддерживающих полный жизненный цикл разработки и отладки моделей, а также проведения экспериментов, и отличающихся следующими функциональными возможностями:

– полным набором функциональных возможностей мультиагентной СДМС;

– проектированием концептуальной модели предметной области;

– возможностью построения мультиагентных моделей, содержащих ИА;

– проблемной ориентацией на процессы преобразования ресурсов;

– интеграцией с методикой BSC;

– поддержкой русского языка.

2. СИТУАЦИОННАЯ МОДЕЛЬ МУЛЬТИАГЕНТНОГО ПРОЦЕССА ПРЕОБРАЗОВАНИЯ РЕСУРСОВ

2.1. Требования к модели

Математическая модель должна обеспечивать динамическое моделирование действий интеллектуальных агентов по анализу ситуаций и принятию решений в области процессов преобразования ресурсов. Таким образом, модель должна отвечать следующим требованиям:

1) ориентация на моделирование процессов преобразования ресурсов и решение соответствующих задач [33] и отражение следующих особенностей данных процессов: учет различных типов ресурсов (материальных, информационных, энергетических, трудовых, финансовых); учет состояния операций и условий в конкретные моменты времени; возможность задания начальных состояний операций; учет возникновения и разрешения конфликтов на общих ресурсах и средствах; возможность задания разветвления и слияния потоков ресурсов; иерархическое представление структуры процесса; возможность расчета характеристик и параметров процесса на каждом уровне иерархии;

2) наличие сообществ интеллектуальных агентов, участвующих в управлении процессом преобразования ресурсов. Данное требование предполагает наличие следующих элементов: модели агента, базы знаний интеллектуального агента, модели взаимодействия (общения) агентов;

3) применение ситуационного подхода (наличия в модели агента механизмов диагностирования ситуаций и поиска решений).

Модель, отвечающая вышеперечисленным требованиям, реализуется с помощью интеграции имитационного, экспертного и ситуационного моделирования в области процессов преобразования ресурсов.

2.2. Основные объекты ситуационной модели процессов преобразования ресурсов, их поведения и отношения

Для предметной области процессов преобразования ресурсов в настоящее время не существует математической ситуационной модели, поэтому разработка соответствующего математического аппарата является актуальной задачей. В данном разделе решается задача расширения модели процессов преобразования ресурсов аппаратом ситуационного управления и мультиагентных систем.

При построении динамических моделей производственных процессов (относящихся к классу процессов преобразования ресурсов) широко используются следующие математические схемы: модели системной динамики [16–17], системы массового обслуживания [15, 74], сети Петри [2, 15], процессы преобразования ресурсов [33–39]. Математическая модель процессов преобразования ресурсов, представленная в работах [33–43], является наиболее близкой для реализации на ее основе ситуационной модели процессов преобразования ресурсов. С целью создания нового математического аппарата в работе за основу берется математическая модель процесса преобразования ресурсов и расширяется аппаратом ситуационного управления и мультиагентных систем. Также в математической модели процесса преобразования ресурсов был усилен аппарат моделирования конфликтов, возникающих на общих ресурсах и средствах.

Для реализации ситуационной мультиагентной модели, ориентированной на динамическое моделирование процессов преобразования ресурсов, как следует из первой главы, в модель [33–43] необходимо ввести следующие объекты: систему; модель системы; цель; команду управления; микроситуацию; макроситуацию; ситуацию; математический формализм; агента, событие.

Понятия системы, модели, микроситуации, макроситуации, ситуации, математического формализма, события заимствованы из работы [18].

Под **системой** (***Sys***) понимается совокупность объектов (элементов), связей между ними и собственных атрибутов. В системе все объекты должны быть связаны. В системе может изменяться значение свойств объектов и связей.

$$Sys = <O, \{Relation\}, A^{self}>, \qquad (2.1)$$

где O – объекты (элементы), ресурсы, средства, сообщения, преобразователи, цели, агенты, параметры;

$Relation$ – связи;

A^{self} – собственные атрибуты системы.

Основными объектами ситуационного мультиагентного процесса преобразования ресурсов являются (рис. 2.1): операции (*Op*), ресурсы (*RES*), команды управления (*U*), средства (*MECH*), процессы (*PR*), источники (*Sender*) и приемники ресурсов (*Receiver*), перекрестки (*Junction*), цели (*G*), параметры (*P*), *агенты* (*Agent*). Отдельно выделены

информационные типы ресурсов: сигналы (*Sig*) и заявки на выполнение операции (*Order*). Параметры процесса задаются функцией от характеристик объектов и разделяются на производные (свертка различного типа характеристик) и консолидированные (свертка одноименных характеристик операций процесса). Описание причинно-следственных связей между элементами преобразования и ресурсами задается объектом «связь» (*Relation*). Агенты управляют объектами процесса преобразования ресурсов и взаимодействуют между собой. Существование агентов предполагает наличие сообщений (*Message*), микроситуаций (*Mis*), макроситуаций (*Mas*), ситуаций (*Sit*) и решений (планов действий) (*Decision*). Элементы с индексом «*Ca*» – условия запуска операций, перекрестков.

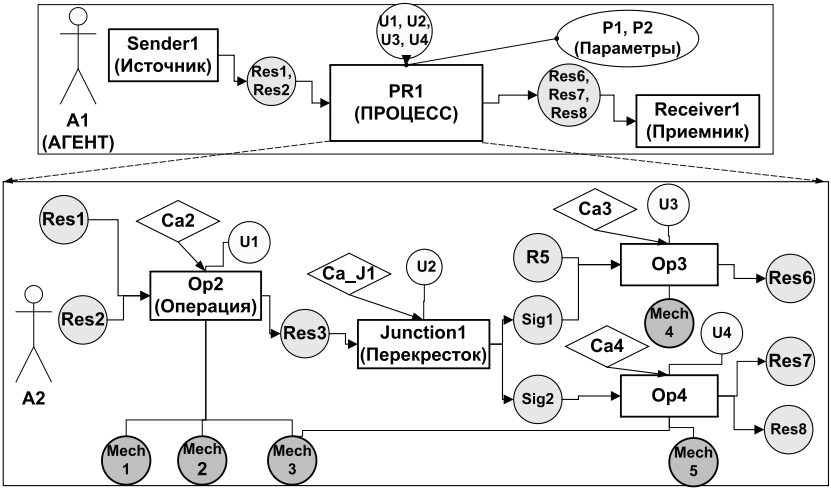


Рис. 2.1. Объекты ситуационного мультиагентного процесса преобразования ресурсов

При описании системы создается ее ***модель*** (***M***). В модели пренебрегают некоторыми связями, параметрами и объектами. Поэтому для одной модели может существовать множество различных систем. Понятие модели будет отличаться от понятия системы набором условий, которые определяют совокупность объектов и связей как систему.

Модели процесса преобразования ресурсов (*M*) соответствует следующая структура:

$$M = <Name,\ desc,\ O,\ \{Relation\},\ A^{self}>, \qquad (2.2)$$

где *Name* – имя модели*;*

desc – описание модели.

Методы: создание объекта процесса преобразования ресурсов; создание агента; создание цели.

Под ***математической формулой*** или ***формализмом*** (***H***), задающим характер взаимоотношений между объектами (правил изменения состояния модели), будем понимать слово, состоящее из последовательности объектов и отношений между ними;

$$H \equiv O_1 Relation\_O_2 Relation_1 O_3 ... Relation_n O_{n+1}, \qquad (2.3)$$

где *Relation_* – отношение эквивалентности.

Возможны следующие варианты: продукционные правила; теоремы; математические механизмы, используемые при рассуждениях с неточными знаниями; формулы; сценарии и т. д. Функция (*F*), которая в дальнейшем будет использоваться при описании объектов мультиагентного процесса преобразования ресурсов, является видом математического формализма.

Цель (*G*). Цели могут быть представлены в различных видах: целевой функции, иерархии (например, И/ИЛИ граф), графа причинно-следственных связей (например, методика BSC, см. раздел 1.2.2), и т. д.

Цель G_j характеризуется: именем цели (*Name*), целевой функцией F_j, зависящей от параметров $\{P^{G(j)}\}$ (в качестве параметров могут выступать атрибуты объектов модели процесса преобразования); текущим значением (*value*)*;* плановым (желаемым) значением (*plan*)*, начальным значением* (*start*)*, критическим значением* (*critical*)*:*

$$G_j = \left\langle Name,\ F_j\left(P_1^{G_j}, ..., P_n^{G_j}\right),\ value, plan, start, critical \right\rangle. \qquad (2.4)$$

Методы: достижение цели; сигнализация критического значения.

Ресурс (*Res*). Каждому ресурсу в модели соответствует следующая структура:

$$RES_b^{Str}(t) = <name, kind, RES_b(t), RES_b^{MAX},$$

$$RES_b^0, RES_b^T, Cost_b, h_{inc}^{res}(t), h_{dec}^{res}(t), Cost_b^{\Sigma}(t), \quad (2.5)$$
$$h_{\Sigma inc}^{res}(t_0, t), h_{\Sigma dec}^{res}(t_0, t), Metric\_R >,$$

где RES_b^{Str} – структура b-го ресурса;

name – имя ресурса;

kind – тип ресурса (материальный, финансовый, информационный, энергетический, трудовой);

$RES_b(t) \in [0, M]$ – текущее значение ресурса;

$RES_b^{MAX} \in [0, M]$ – максимально возможное значение ресурса;

RES_b^0 – начальное значение ресурса;

RES_b^T – конечное значение ресурса;

$Cost_b$ – цена единицы b-го ресурса;

$h_{inc}^{res}(t)$ – приращение в текущий момент времени;

$h_{dec}^{res}(t)$ уменьшение в текущий момент времени;

$Cost_b^{\Sigma}(t) = RES_b(t)Cost_b$ – стоимость b-го ресурса;

t_0 – время начала моделирования;

$h_{\Sigma inc}^{res}(t_0, t)$ – суммарное приращение ресурса за интервал времени;

$h_{\Sigma dec}^{res}(t_0, t)$ – суммарное уменьшение ресурса за интервал времени.

Metric_R – единица измерения ресурса.

Над ресурсом могут выполняться следующие операции, – в терминологии объектно-ориентированного подхода – (ООП) методы: уменьшение ресурса, увеличение ресурса, анализ состояния ресурса.

Перед рассмотрением объекта **средство** введем понятия события и действия. Под простейшим **событием (e^{τ})** понимается момент времени, в который произошло изменение значения какого-либо атрибута модели.

$$e^t = \left\langle t,\ a,\ z_1^a,\ z_2^a \right\rangle;\ \ z_1^a \neq z_2^a, \quad (2.6)$$

где τ – момент времени;

a – атрибут;

z_1^a – значение атрибута до события;

z_2^a – значение атрибута после события.

Под **действием** (***A***) понимается изменение состояний объекта или модели, происходящее за конечный промежуток времени. Действие

характеризуется событием и функцией действия, приводящей к изменению значения атрибута объектов.

$$A = <e, F^A, \{a\}>, \tag{2.7}$$

где e – событие;

F^A – функция действия;

$\{a\}$ – множество изменяемых атрибутов.

Средство (*Mech*). Каждому средству в модели соответствует следующая структура:

$$\begin{gathered} MECH_y^{Str} = \Big\langle name, kind, MECH_y(t), MECH_y^{ALL}, t_{Create}^{mech}, \\ Status_{Mech}, Am^{in}, Am^{out}, Am^{use}, Am^{lock}, Am^{unlock}, Am^{other}, \\ res_y^{in}, res_y^{out}, res_y^{use}, res_y^{lock}, res_y^{unlock}, res_y^{other}, t_y^{other}, \\ product_y, Cost_y, t_{\sum use}^{mech}(t), t_{\sum s\tan ding\_idle}^{mech}(t) \Big\rangle, \end{gathered} \tag{2.8}$$

где $MECH_y^{Str}$ – структура y-го средства;

name – имя средства;

kind – тип средства (оборудование / трудовой ресурс);

$MECH_y(t) \in [0, M]$ – текущее количество свободных y-х средств;

$MECH^{ALL}_y \in [0, M]$ – всего средств;

t_{Create}^{mech} – время создания;

StatusMech = {*wait*, *active*, *lock*, *other*} – состояние средства, определенное на конечном множестве состояний: *wait* – ожидание, *active* – выполнение, *lock* – прерывание, *other* – набор расширяемых состояний, который может быть связан с различными видами возникаемых поломок и действий по их устранению;

Am^{in} – действие по запуску средства в момент начала преобразования;

Am^{out} – действие по остановке средства в момент окончания преобразования;

Am^{use} – действие по выполнению преобразования;

Am^{lock} – действие по остановке средства в момент прерывания преобразования;

Am^{unlock} – действие по запуску средства в момент продолжения преобразования;

Am^{other}– действие по устранению поломки;

res_y^{in} – единоразовые затраты ресурсов при начале преобразования (запуск средства);

res_y^{out} – единоразовые затраты ресурсов при окончании преобразования (остановка средства);

res_y^{use} – расход ресурсов в единицу времени, обеспечивающий работу средства, может быть задан функцией;

res_y^{lock} – единоразовые затраты ресурсов при захвате средства другой операцией (при блокировке текущей операции);

res_y^{unlock} – единоразовые затраты ресурсов при освобождении средства (продолжении преобразования после прерывания);

res_y^{other} – затраты ресурсов при возникновении и устранении поломки;

t_y^{other} – периодичность возникновения поломки, может быть задана функцией;

$product_y$ – производительность средства в единицу времени, может быть задана функцией от расхода ресурсов;

$Cost_y$ – начальная цена единицы y-го средства;

$t_{\sum use}^{mech}(t)$ – суммарное время использования средства;

$t_{\sum standing\_idle}^{mech}(t)$ – суммарное время простоя средства.

С точки зрения удобства описания модели реальных процессов преобразования ресурсов и усиления аппарата конфликтов в рамках данной работы была введена модель поведения средства (действия – Am^{in}, Am^{out}, Am^{use}, Am^{lock}, Am^{unlock}, Am^{other}) и расширен наборатрибутов (t_{Create}^{mech}; $Status_{Mech}$; res_y^{in}; res_y^{out}; res_y^{use}; res_y^{lock}; res_y^{unlock}; res_y^{other}; t_y^{other}; $product_y$).

Выполнение любого действия может приводить как к потреблению ресурсов y-м средством, так и восполнению (в этом случае знак «минус» меняется на «плюс»):

$$Am^k : \forall j, RES_j^k(t) = RES_j^k(t) - res_y^k(t). \qquad (2.9)$$

Над средствами выполняются следующие методы: создание, удаление, захват, освобождение, амортизация, отказ (поломка), восстановление.

Операция (*Op*). Процесс преобразования ресурсов может состоять из одной или множества операций. Основа элементарной операции процесса преобразования ресурсов (Op_k) была взята из [33–37] и [118]

и была дополнена в направлении ситуационного управления, а также усиления аппарата моделирования конфликтов, возникающих на общих ресурсах и средствах. Op_k соответствует следующая структура:

$$Op_k = <f,\ in,\ out,\ u,\ res_{Lock},\ res_{UnLock},\ ch_{Op},\ g_{Op},\ c_a, mech,$$

$$Status_{Op},\ time,\ prior,\ kind\_prior,\ break\_off>, \qquad (2.10)$$

где f – функция, реализуемая операцией;
$in = \{in_1,...,in_n\}$ – множество входов, различных типов;
$out = \{out_1,...,out_m\}$ – множество выходов, $out = f(in)$;
$u = \{u_1, ..., u_z\}$ – множество команд управления;
$res_{Lock} = \left\{res^1_{Lock},\ \dots\ res^z_{Lock}\right\}$, – множество ресурсов, необходимых для прерывания операции, $res_{UnLock} = \left\{res^1_{UnLock},\ \dots, res^w_{UnLock}\right\}$ – множество ресурсов, необходимых для продолжения выполнения операции;
$ch_{Op} = \left\{ch^1_{Op},\dots, ch^k_{Op}\right\}$ – характеристики операции;
g_{Op} – цели операции (требуемые значения характеристик операции), $g_{Op} \subseteq ch_{Op}$;
c_a – условие запуска операции;
$mech = \{mech_1,...,mech_q\}$ – средства преобразования;
$Status_{Op} = \{wait, active, lock, done\}$ – состояние операции, определенное на конечном множестве состояний: *wait* – ожидание, *active* – выполнение, *lock* – прерывание, *done – выполнена;*
time – длительность выполнения преобразования;
prior – приоритет операции задает очередность выполнения операций, может быть описан постоянной величиной или функцией, т. е. быть статическим или динамическим;
kind_prior – тип приоритета (относительный, абсолютный);
$break\_off = \{true, false\}$ –признак запрета прерывания; если «*true*» – правило не может прерываться.

В случае остановки выполнения операции по каждому входному ресурсу (*in*) устанавливается признак $stop\_los = \{true, false\}$ – признак потери ресурса. Если $stop\_los = false$, то захваченное количество ресурса для обработки при остановке операции возвращается на склад.

Предложенная ранее модель операции в [33] не позволяет учитывать следующие особенности реальных процессов преобразования

ресурсов при прерывании выполнения одной операции (Op_i) другой операцией (Op_j):

1) затраты ресурсов, включая затраты времени на проведение подготовительных операций, необходимые на осуществление остановки (прерывания) операции Op_i(см. формулу (2.8));

2) затраты ресурсов, включая затраты времени, необходимые на продолжение операции (Op_i) (после выхода из состояния прерывания).

Так, например, при поступлении срочного заказа на производство, выполнение менее срочного заказа на заводе «замораживается», но прерывание запущенных операций: во-первых; не происходит мгновенно; во-вторых, требует дополнительных ресурсов. При продолжении выполнения «замороженного» заказа требуется дополнительное время и ресурсы на переналадку оборудования и подготовительные операции.

Условие запуска (c_a) задается следующим образом:

$$C_a(t) = C_a^{in}(t) \wedge C_a^{out}(t) \wedge C_a^{u} \wedge C_a^{mech}(t) \wedge C_a^{status}(t) \wedge C_a^{time}(t), \qquad (2.11)$$

где C_a^{in} – условие наличия необходимых входных ресурсов;

C_a^{out} – условие учета ограничений выхода;

C_a^{u} – **условие наличия разрешающих команд управления;**

C_a^{mech} – условие готовности необходимых средств;

C_a^{status} – условие готовности к исполнению;

C_a^{time} – условие запуска по времени.

Переход операции в состояние «выполнение» сопровождается выполнением действий по захвату входных ресурсов A_{in}^{RES} и средств A_{in}^{MECH}. Находясь в состоянии «выполнение», операция может перейти в состояние «прерывание». Операция может быть прервана для того, чтобы обеспечить выполнение другой операции. Переходя в состояние «прерывание», операция запоминает момент остановки и освобождает захваченные средства A_{Lock}^{MECH}.

В течение состояния «прерывания» проверяется условие наличия свободных средств $C_a^{mech}(t)$. Операция находится в состоянии «прерывания» до тех пор, пока не освободятся необходимые средства. В случае $C_a^{mech}(t) = true$ операция переходит в состояние «выполнение»: захватываются средства A_{UnLock}^{MECH} и продолжается выполнение.

Операция находится в состоянии «выполнение» до тех пор пока $t < t_{End}^{k}$, где t_{End}^{k} – момент окончания k-й операции. При выполнении

условия $t = t_{E}^{k}$, операция переходит в состояние «ожидание». Данный переход сопровождается действиями по формированию выходных ресурсов A_{out}^{RES} и освобождением захваченных средств A_{out}^{MECH}.

Источники и приемники ресурсов. Источники ресурсов (***Sender***) – элементы, моделирующие входные воздействия внешней среды (элементы только с выходами). Приемники ресурсов (***Receiver***) – элементы, моделирующие выходы во внешнюю среду (элементы только с входами). Объекты источники и приемники ресурсов имеют структуру, которая отличается от операции отсутствием средств и действий, выполняемых над средствами.

Перекресток (***Junction***) – элемент, описывающий слияние и разветвление процессов. Активизацией для перекрестков является поступление сообщения от входящих процессов, завершивших свою работу. При положительном решении перекресток генерирует сообщение, разрешающее работу следующих за ним процессов. Базовый класс ***«перекресток»*** (рис. 2.2) имеет следующую структуру:

$$Junction_l = \left\langle Name_l, Message_{IN}, Message_{OUT}, c_a^l, in_l, out_l, f_l, t_{sync}, Aj \right\rangle, \quad (2.12)$$

где $Name_l$ – имя перекрестка;
$Message_{IN}$ – входное сообщение;
$Message_{OUT}$ – выходное сообщение;
c_a^l – условие запуска;
in_l – имена входов информационных;
out_l – имена выходов;
f_l – функция решения;
t_{sync} – время синхронизации;
Aj – модель поведения (действий) перекрестка.

За основу системы действий Aj перекрестков взяты действия операции со следующими изменениями:

– нет захвата / освобождения / блокировки / снятия блокировки средств;

– добавлено $Action_{Junction}^{Z}$, введенное для определения выходов, на которых будут сформированы сигналы (перекрестки ветвления «ИЛИ» и «Исключающее ИЛИ»).

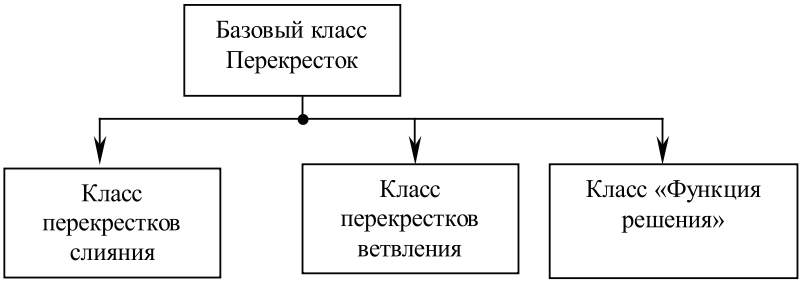


Рис. 2.2. Иерархия классов перекрестков

Сообщению/Команде соответствует следующая структура:

$$Message = <name, kind, Msender, Mresiver, text, prior, read, t_{create}, t_{wait}>, \quad (2.13)$$

где *name* – имя сообщения;
Kind – тип (команда, сообщение);
Msender – отправитель;
Mresiver – получатель;
Text – текст сообщения;
Prior – приоритет;
read = {*true, false*} – признак обработки (чтения сообщения);
t_{create} – время создания;
t_{wait} – время ожидания в очереди.

Методы: создание сообщения/команды, обработка сообщения/команды, удаление сообщения/команды.

Команды управления определяют множество процессов и операций (множество может быть вырождено), когда и в какой последовательности должны быть запущены или остановлены. Команда управления содержит параметры (размещаются в свойстве *text*), определяющие характеристики процесса/операции (например, длительность операции, количество полуфабрикатов или продуктов, которые необходимо произвести, и т. д.).

Каждой **заявке** соответствует следующая структура:

$$Order_j = <name, count, real, lock, owner, parent, prior, t_{create}, t_{wait}>, \quad (2.14)$$

где *name* – имя заявки *j*-го типа;
count – заказываемый объем работ *j*-го типа;

real – выполненный объем работ *j*-го типа;

lock = {*true, false*} – признак блокировки заявки, устанавливается в "true" на время обработки операцией, «источником», «приемником», агентом;

owner – если lock = "true", то это имя элемента, обрабатывающего заявку, иначе – имя последнего блока, обработавшего заявку;

parent – имя блока, создавшего заявку;

prior – приоритет заявки;

t_{create} – время создания заявки;

t_{wait} – время ожидания заявки в очереди.

Учитывая динамическую природу заявок, можно определить **очередь**:

$$\begin{aligned} &Queue_k^r(t) = \left\{ Order_j^1, ..., Order_j^l, ..., Order_j^m \right\}, \\ &\forall Order_j^l : (owner = Op_r) \wedge \left(lock = "false"\right), \end{aligned} \tag{2.15}$$

где $Queue_k^r$ – *k*-я очередь на выходе *r*-й операции;

$Order_j^l$ – заявки в очереди.

К заявкам применимы следующие операции: включение заявки в очередь, упорядочение заявок, исполнение (обработка), исключение заявки из очереди.

Каждому **сигналу** соответствует следующая структура:

$$Sig_d^{Str}(t) = \left\langle name, Sig_d(t), Sig_d^{MAX} \right\rangle, \tag{2.16}$$

где Sig_d^{Str} – структура *d*-го сигнала;

name – имя сигнала;

$Sig_d(t)$ – текущее количество сигналов;

Sig_d^{MAX} – максимально возможное количество сигналов.

Над сигналами выполняются следующие операции: включение в множество сигналов; обработка сигнала; исключение из множества сигналов.

Процесс. Процессы (***PR***) – элементы с входами и выходами. Процесс может состоять из операций, перекрестков и процессов, ниже приводится структура процесса:

$$PR_p = <f, in, out, u, ch_{PR}, g_{PR}, c_a, mech, Status_{PR}, prior_{PR}, s, r_{PR}>, \quad (2.17)$$

где f – функция, реализуемая процессом;
$in = \{in_1,...,in_n\}$ – множество входов, различных типов;
$out = \{out_1,...,out_m\}$ – множество выходов, $out = f(in)$;
$u = \{u_1, ..., u_z\}$ – множество команд управления;
ch_{PR} – характеристики процесса представляют собой множество измеряемых параметров процесса;
$g_{PR} = opt(ch_{PR})$ – цель процесса преобразования – желаемая величина характеристики;
c_a – условие запуска;
$mech = \{mech_1,...,mech_q\}$ – средства преобразования;
$Status_{PR}$ – состояние процесса;
$prior_{PR}$ – приоритет процесса;
s – состав процесса представлен множеством других процессов, называемых в этом случае подпроцессами. Совместное взаимодействие этих подпроцессов реализует функцию данного процесса;
r_{PR} – связи определяют все формы взаимодействия входов и выходов с подпроцессами.

Параметр (*P*). Каждому параметру процесса преобразования ресурсов соответствует структура:

$$P_d = <P_d(t), P_d(t_0), name, desc, plan, f_P>, \quad (2.18)$$

где $P_d(t)$ – текущее значение параметра;
$P_d(t_0)$ – начальное значение параметра;
$name$ – имя параметра;
$desc$ – описание параметра;
$plan$ – плановое значение;
f_P – функция вычисления параметра.

Связь (***Relation***). Для описания причинно-следственных связей (отношений) между вышеперечисленными элементами дополнительно используется класс связь. Структура элемента «связь»:

$$Relation_{AB} = <Name, desc, Element_A, Element_B>, \quad (2.19)$$

где *Name* – имя связи (так для большинства элементов модели ресурсов и преобразователей типовыми будут связи «является входом», «является выходом», «является средством» и т. д.);

desc – описание связи;

$Element_A$, $Element_B$ – имена двух элементов модели процесса преобразования, между которыми установлена связь.

Микроситуации (***Mis***) соответствуют выделенные состояния объекта, которые описываются не значениями каждого атрибута, а символичной записью в виде одного слова или небольшого фрагмента естественно-языкового описания. Микроситуация может соответствовать нескольким состояниям объекта, т. е. являться особым макросостоянием. Особенность микроситуации заключается в том, что она может зависеть не от всего сектора параметров объекта, а от его части (проекции). В общем случае микроситуация определяется с помощью рассуждений (правил) ЭС.

$$Mis = <name, desc, \{a\}, \{Z^a\}, h(\{a\}, \{Z^a\})>, \quad (2.20)$$

где *name* – имя микроситуации;

desc – описание микроситуации;

$\{a\}$ – множество атрибутов объекта, определяющее микроситуацию;

$\{Z^a\}$ – диапазон значении атрибутов, допустимых для заданного типа микроситуации;

$h(\{a\},\{Z^a\})$ – правила определения микроситуации.

Ситуации (***Sit***) *соответствует* совокупность микроситуаций и (или) значений отдельных свойств объектов и некоторых произошедших событий. Включение событий в определение ситуации обусловлено тем, что ситуация может складываться (накапливаться, нагнетаться) в течение времени. Ситуация также может сохраняться в течение какого-то времени. Эта временная зависимость выражается в произошедших событиях.

Правила, определяющие ситуацию, могут зависеть от непроизошедших событий. Например, если ни в одном объекте не возникло экстремальных микроситуаций, то в целом ситуация стабильна.

$$Sit = <name,\ desc,\ \{Mis\},\{a\},\{e\},\ h(\{Mis\},\{a\},\{e\})>, \qquad (2.21)$$

где *name* – имя ситуации;
desc – описание ситуации;
{*Mis*} – множество микроситуаций;
{*a*} – множество атрибутов объектов;
{*e*} – множество событий;
h({*Mis*},{*a*},{*e*}) – правила определения ситуации.

Макроситуации (***Mas***) соответствует совокупность всех ситуаций, микроситуаций и произошедших событий в модели. Макроситуация необходима для оценки системы (модели) в целом.

$$Mas = <name,\ desc,\ \{Mis\},\{Sit\},\{e\},\ h(\{Mis\},\{Sit\},\{e\})>, \qquad (2.22)$$

где *name* – имя макроситуации;
desc – описание макроситуации;
{*Mis*} – множество микроситуаций;
{*Sit*} – множество ситуаций;
{*e*} – множество событий;
h({*Mis*},{*Sit*},{*e*}) – правила определения макроситуации.

Решению (*Decision*) соответствует следующая структура:

$$Decision = <Name,\ desc,\ A\_Decision>, \qquad (2.23)$$

где *Name* – имя решения;
desc – описание решения;
A_Decision – описание решения в виде параллельно-последовательных во времени действий и команд управления элементами процесса преобразования ресурсов, выполнение которых приведет к переходу из ситуации Sit_0 в Sit_k.

Агент (Ag) имеет следующую структуру:

$$Agent = <Name,\ G\_Ag,\ prior,\ KB\_Ag,\ Mess\_In\_Count,\ Mess\_Out\_Count,\ SPA,\ Control\_Objects,\ AU,\ AD>, \qquad (2.24)$$

где *Name* – имя агента;
G_Ag – цели агента;

prior – приоритет агента;
KB_Ag – база знаний агента (знания агента, know ledgebase);
Mess_In_Count – количество входящих сообщений;
Mess_Out_Count – количество исходящих сообщений;
SPA – сценарии поведения;
Control_Objects – множество управляемых объектов процесса преобразования ресурсов;
AU – множество агентов «начальников», может быть вырождено;
AD – множество агентов подчиненных, может быть вырождено.

Методы агента: анализ мира (*World_analyzes*); диагностирование ситуации (*Situations_diagnostic*); поиск решения (*Decision_support*); чтение сообщений (*R_Mess*); написание сообщений (*W_Mess*).

Агенты управляют объектами процесса преобразования ресурсов. Агент выполняет следующие действия:

- анализирует окружение (текущую ситуацию);
- диагностирует ситуацию, обращается к базе знаний. В случае определения некоторой соответствующей ситуации (исключительной) агент пытается найти решение (сценарий действий) в базе знаний или выработать его самостоятельно;
- вырабатывает (принимает) решение;
- определяет (переопределяет) цели;
- контролирует достижение целей;
- делегирует цели своим и чужим объектам процесса преобразования ресурсов, а также другим агентам;
- обменивается сообщениями.

Элементы процесса преобразования ресурсов участвуют в обмене сообщениями и на основе своих моделей поведения выполняют свои преобразовательные функции, руководствуясь поступающими сообщениями.

Диаграмма прецедентов, определяющая отношения между агентом и элементом процесса преобразования ресурсов, представлена на рис. 2.3. Диаграмма прецедентов (англ. use case – вариант использования) представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом) [100]. Действующим лицом

может выступать некое устройство. Диаграмма прецедентов – одна из диаграмм языка UML [99–100, 102–105].

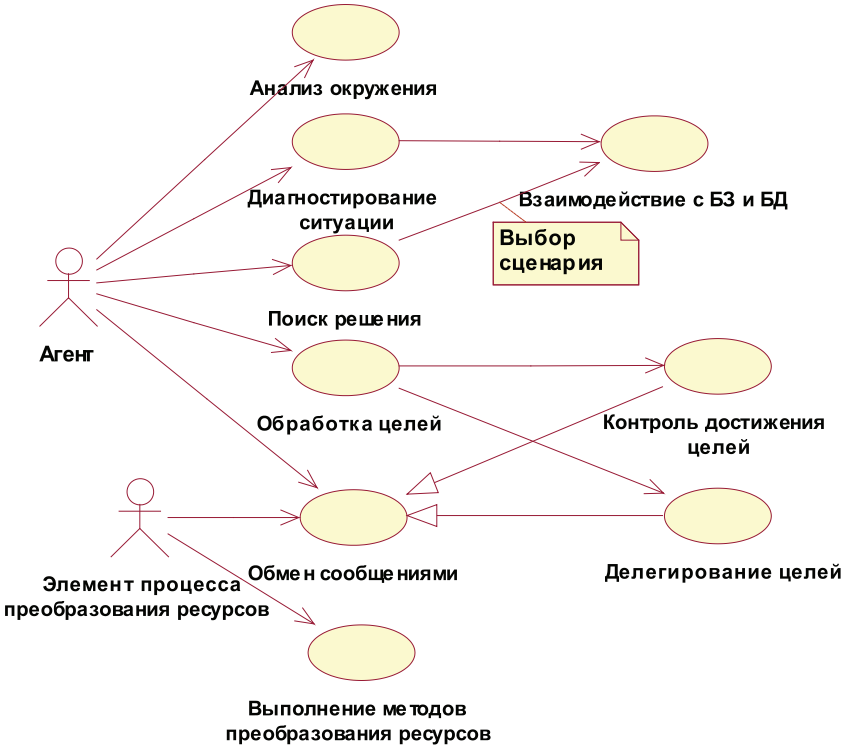


Рис. 2.3. Диаграмма прецедентов, определяющая отношения между агентом и элементом процесса преобразования ресурсов

Для описания иерархической структуры моделируемой системы (процесса преобразования ресурсов) использован аппарат системных графов высокого уровня интеграции [2–3, 33–43]:

$$\overrightarrow{PR}^{\Sigma}_{L=i} = \left\langle \left\{ Sender^{m} \cup Op^{m} \cup Receiver^{m} \cup Junction^{m} \cup Agent^{m} \right\}_{L=i}; \right. \\ \left. \left\{ PR^{p_i}_{L=j}; p_i = 1,\ldots,n^{p}_{L=j} \right\}_{j=2,\ldots,i}; \left\{ Relation^{mk}_{AB} \right\}_{L=i} \right\rangle. \quad (2.25)$$

Граф i-го уровня интеграции образуется в результате поэтапной интеграции графов $\overrightarrow{PR}_1^{\Sigma}, \overrightarrow{PR}_2^{\Sigma}$ $\overrightarrow{PR}_{i-1}^{\Sigma}$,..., с образованием на каждом j-м этапе множества $\left\{PR_{L=j}^{p}; p=1,\ldots,n_{L=j}^{p}\right\}$ процессов (подпроцессов) j-го уровня интеграции, L – уровень интеграции. Элементы множества мультиагентного процесса преобразования ресурсов

$$\{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=I} \subset$$
$$\subset \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=i\text{-}1} \subset \ldots$$
$$\ldots \subset \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}$$

и множества ресурсных отношений

$$\left\{Relation_{AB}^{mk}\right\}_{L=i} \subset \left\{Relation_{AB}^{mk}\right\}_{L=i-1} \subset \ldots \subset \left\{Relation_{AB}^{mk}\right\}$$

системного графа $\overrightarrow{PR}_{L=i}^{\Sigma}$ представляют собой элементы процесса преобразования и ресурсные отношения между элементами, а также элементы

$$Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m$$

и ресурсные отношения $Relation_{AB}^{mk}$ системного графа $\overrightarrow{PR}^{\Sigma}$ нулевого уровня интеграции, не вошедшие при поэтапной интеграции ни в один процесс $PR_{L=j}^{p}$.

Каждая вершина системного графа процесса преобразования ресурсов [33–43] характеризуется некоторым набором атрибутов (показателей) $h_1, \ldots, h_z$. Все множество атрибутов вершин системного графа задает атрибутивное множество. Вычисление интегральных показателей процессов (системных вершин) $h_1, \ldots, h_z$ на произвольном i-м уровне ($i > 0$) задается над вершинами (i –1)-го уровня интеграции.

Иерархическое представление ситуационно-агентной модели процесса преобразования ресурсов в виде системного графа высокого уровня интеграции представлено на рис. 2.4.

Для построения ядра моделирующей системы в [33] был использован аппарат продукционных систем [11, 92, 119]. С учетом расширений модели определим структуру продукционной системы мультиагентного процесса преобразования ресурсов:

$$PS = < Rps, Bps, Ips >, \qquad (2.26)$$

где $Rps = \{RES(t)\} \cup \{MECH(t)\} \cup \{Order(t)\} \cup \{Sig(t)\} \cup \{Message(t)\} \cup$ $\cup \{U(t)\} \cup \{G(t)\}$ – текущее состояние ресурсов, средств, заявок, сигналов, сообщений, команд управления, целей (рабочая память);

Bps – множество правил преобразования ресурсов и действий агентов (база знаний);

Ips – машина вывода, состоящая из планировщика и машины логического вывода по БЗ агентов).

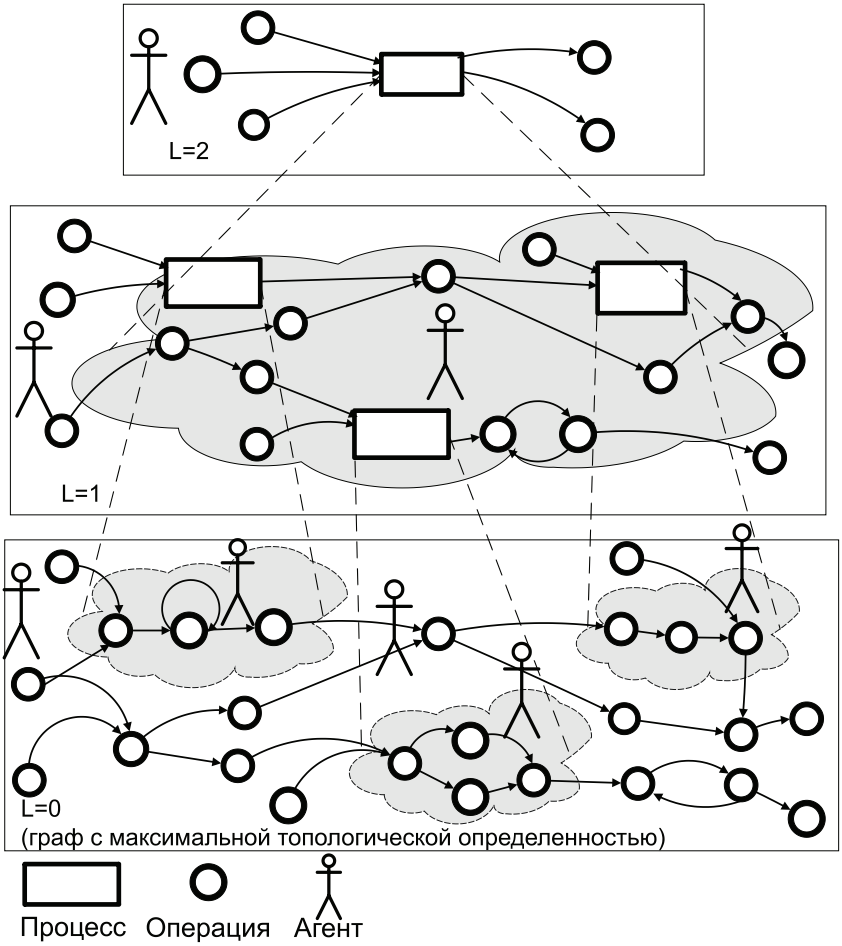


Рис. 2.4. Иерархическое представление процессов преобразования ресурсов и агентов

Определена структура правила преобразования, которая соответствует структуре операции процесса преобразования ресурсов (Op_k):

$$RULE_k^{OP} = < C_a(t),\ A_{IN}(t_{Ca}),\ A_{Lock}(t_{Lock}),\ A_{UnLock}(t_{UnLock}),\ A_{OUT}(t_{End}),$$
$$Status^{RULE},\ time_{RULE},\ prior,\ kind\_prior,\ break\_off>; \quad (2.27)$$

$$A_{IN}\left(t_{Ca}\right) = \left\langle A_{in}^{RES}\left(t_{Ca}\right),\ A_{in}^{MECH}\left(t_{Ca}\right)\right\rangle; \quad (2.28)$$

$$A_{Lock}\left(t_{Lock}\right) = \left\langle A_{Lock}^{MECH}\left(t_{Lock}\right), A_{Lock}^{RES}\left(t_{Lock}\right)\right\rangle; \quad (2.29)$$

$$A_{UnLock}\left(t_{UnLock}\right) = \left\langle A_{UnLock}^{RES}\left(t_{UnLock}\right),\ A_{UnLock}^{MECH}\left(t_{UnLock}\right)\right\rangle; \quad (2.30)$$

$$A_{OUT}\left(t_{End}\right) = \left\langle A_{in}^{RES}\left(t_{End}\right),\ A_{in}^{MECH}\left(t_{End}\right)\right\rangle, \quad (2.31)$$

где $C_a(t)$ – условие запуска правила;
$A_{IN}(t_{Ca})$ – действия по захвату входных ресурсов ($A_{in}^{RES}(t_{Ca})$) и захвату средств $A_{in}^{MECH}\left(t_{Ca}\right)$;
$A_{Lock}(t_{Lock})$ – действия по прерыванию операции: освобождению захваченных средств $\left(A_{Lock}^{MECH}\left(t_{Lock}\right)\right)$, которое может сопровождаться соответствующими затратами ресурсов $\left(A_{Lock}^{RES}\left(t_{Lock}\right)\right)$ – затраты ресурсов, включая ресурс времени, необходимые для выполнения остановки операции Op_i);
$A_{UnLock}(t_{UnLock})$ – действия по продолжению выполнения операции: захвату свободных средств $\left(A_{UnLock}^{MECH}\left(t_{UnLock}\right)\right)$, которое может сопровождаться соответствующими затратами ресурсов $\left(A_{UnLock}^{RES}\left(t_{UnLock}\right)\right)$ – затраты ресурсов, включая ресурс времени, необходимые для продолжения выполнения операции Op_i);
$A_{OUT}(t_{End})$ – действия по формированию выходных ресурсов $\left(A_{in}^{RES}\left(t_{End}\right)\right)$ и освобождению средств $\left(A_{in}^{MECH}\left(t_{End}\right)\right)$;
$Status^{RULE}$ = {*wait, active, lock, done*} – состояние правила, определенное на конечном множестве состояний: *wait* – ожидание, *active* – выполнение, *lock* – прерывание, *done* – выполнено;
$time_{RULE}$ – длительность выполнения правила преобразования;
prior – приоритет правила;
kind_prior – тип приоритета (относительный, абсолютный);
break_off = {*true, false*} – признак запрета прерывания; если «*true*» – правило не может прерываться.

К правилам преобразования дополнительно, в отличие от модели в [33], относятся также средства, обладающие моделью поведения. Каждому средству $MECH_y$, обладающему моделью поведения, соответствует следующая структура правила:

$$RULE_y^{MECH} = \left\langle t_{Create}^{mech}, Status_{Mech}, Am^{in}, Am^{out}, \right.$$
$$\left. Am^{use}, Am^{lock}, Am^{unlock}, Am^{other}, t_y^{other}, product_y \right\rangle, \quad (2.32)$$

где t_{Create}^{mech} – время создания;

$Status_{Mach}$ = {*wait, active, lock, other*} – состояние средства, определенное на конечном множестве состояний: *wait* – ожидание, *active* – выполнение, *lock* – прерывание, *other* – набор расширяемых состояний, который может быть связан с различными видами возникаемых поломок и действий по их устранению;

Am^{in} – действие по запуску средства в момент начала преобразования;

Am^{out} – действие по остановке средства в момент окончания преобразования;

Am^{use} – действие по выполнению преобразования;

Am^{lock} – действие по остановке средства в момент прерывания преобразования;

Am^{unlock} – действие по запуску средства в момент продолжения преобразования;

Am^{other} – действие по устранению поломки;

t_y^{other} – периодичность возникновения поломки, может быть задана функцией;

$product_y$ – производительность средства в единицу времени, может быть задана функцией от расхода ресурсов.

Алгоритм работы машины вывода состоит из следующих основных этапов: определение текущего момента времени $SysTime = \min\limits_{j \in RULE} T_j$; формирование очереди правил преобразования; выполнение правил преобразования и изменение состояния рабочей памяти (ресурсов и средств).

Чтобы представить работу ситуационной модели, рассмотрим модель Ю.И. Клыкова.

2.2.1. Анализ работы классической ситуационной (семиотической) модели

Краткое описание классической ситуационной (семиотической) модели приводится на основе [8]. Механизм работы ситуационной (семиотической) модели, представленный Ю.И. Клыковым, уже является «классическим» и предполагает разбиение на следующие составляющие:

– M_1 – имитационная микромодель структуры и законов функционирования объекта управления (дискретная сеть модели M_1 описывает модель динамического процесса);

– M_2 – имитационная модель процесса формирования обобщенного управления большой системой (модель ситуационной (семиотической) системы M_2), которая состоит из следующих подмоделей: анализа ситуаций M_{21}; структурирования (корреляции) ситуации M_{22}; обобщения ситуаций M_{23}; экстраполяции ситуаций M_{24};

– M_3 – имитационная макромодель управления объектом.

Переход сети из состояния $s(t)$ в состояние $s(t + 1)$ определяется в этом случае с помощью подстановок вида $\varphi \rightarrow \psi$, возникающих в результате обобщения ситуаций-решений, где φ – фрагмент, удаляемый из ситуации $s(t)$, а ψ – фрагмент, включаемый в $s(t)$ вместо φ. В результате реализации подстановки $\varphi \rightarrow \psi$ образуется ситуация $s(t + 1)$. Последовательность преобразований, реализующая подстановку $\varphi \rightarrow \psi$, задается с помощью дискретной сети. Конечное непустое множество подстановок вида $\varphi \rightarrow \psi$, для которого определен способ реализации подстановок на дискретной сети, образует трансформационную грамматику. Время τ реализации подстановки $\varphi \xrightarrow{\tau}$ имитирует в соответствующем масштабе длительность существования отношений в объекте управления, задаваемых с помощью подстановки [8]. *Таким образом, время τ (время перехода из одной ситуации в другую) эквивалентно реализации системного времени (SysTime) в имитационном моделировании.*

Управление большой системой с помощью модели M_3 показано на следующей структурной схеме (рис. 2.5).

Для моделирования процессов среда моделирования должна вести учет времени, протекающего в модели. Это время будем называть модельным. Можно по-разному организовывать учет времени и продвижение процессов в модели. В [138] приведены два следующих подхода учета модельного времени.

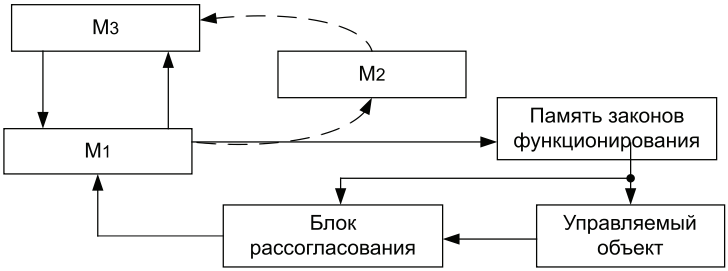


Рис. 2.5. Управление большой системой

Последовательный просмотр всеми процессами моментов времени с фиксированным шагом, приравненным некоторому интервалу в реальной системе, является наиболее простым. Такой подход достаточно общий и позволяет моделировать и непрерывные динамические звенья. В этом случае в среде моделирования существует системный процесс – модельный таймер, который последовательно увеличивает значение внутренней переменной (например, глобальной переменной *SysTime*) и оповещает все остальные процессы модели о сделанных изменениях. Другие процессы в этом случае синхронизируются в соответствии со значениями доступного им всем модельного времени.

Использование централизованного календаря событий является альтернативным методом управления событиями процессов. В этом случае должен существовать процесс-монитор, внутри которого ведется список всех процессов с указанием их состояний. Различаются два типа состояния ожидания процесса: ожидание фиксированного момента времени и ожидание некоторого асинхронного события. В первом случае монитор активизирует процесс по достижении требуемого значения модельного времени. Таймер модельный можно сразу устанавливать в ближайшее время активизации среди всех ожидающих в этом состоянии процессов для сокращения накладных расходов на перебор моментов времени, то есть значение *SysTime* можно положить равным:

$$SysTime = \min_{j \in PR} T_j,$$

где T_j – время активизации j-го процесса;

PR – множество ожидающих истечения интервала времени процессов.

Так как асинхронные события, которые ожидает вторая группа процессов, смогут появиться только в результате срабатывания процессов первой группы, то все процессы второй группы должны активизироваться для проверки своих условий после обработки всех процессов первой группы, запланированных на данный момент.

Таким образом, алгоритм работы машины вывода, реализующий ситуационный подход к проблемной области процессов преобразования ресурсов, должен состоять из следующих основных этапов: определение текущего момента времени $SysTime = \min_{j \in PR} T_j$; *диагностирование текущих ситуаций, выработка команд управления (корректировка управляющих воздействий), формирование очереди правил преобразования; выполнение правил преобразования и изменение состояния рабочей памяти (ресурсов и средств).*

2.2.2. Анализ SIE-модели А.Ю. Филипповича

Интегрированная ситуационная, имитационная, экспертная модель А.Ю. Филипповича (SIE-модель) представлена в работе [18]. В силу того, что данная модель ориентирована на проблемную область допечатных процессов (полиграфии), отдельные ее фрагменты изложим в терминах модели процессов преобразования ресурсов.

SIE-модель представлена в виде нескольких различных уровней, соответствующих имитационному, экспертному и ситуационному представлению информации. Общие элементы либо дублируются (проецируются) на каждом уровне в различном представлении, либо содержатся в одном уровне. В последнем случае уровни имеют возможность обратиться к соответствующему элементу [18].

Первый уровень модели предназначен для описания структуры системы. Для этого каждому объекту (субъекту) сопоставляется блок. Все блоки соединяются между собой каналами взаимодействия. По этим каналам могут перемещаться динамические объекты (транзакты). Каждый блок определенное время обрабатывает транзакт и задерживает его на время, которое определяется интенсивностью работы устройства. Блоки не изменяют характеристики транзактов. Большинство блоков SIE-модели имеют свои аналоги в известных СИМ (AnyLogic, Arena, GPSS, Simio) и могут быть легко преобразованы.

В ситуационной модели мультиагентного процесса преобразования ресурсов (МППР) транзактам и блокам SIE-модели могут

соответствовать соответственно ресурсы (*Res, Order, Message*) и преобразователи (*Op, PR, Junction, Sender, Resiver*), с учетом того, что они могут изменять характеристики транзактов.

На структурном уровне не существует возможности задания произвольных событий, сцепленных процессов, условных операторов и программ по изменению структуры модели. Для выполнения условных операторов используется специальный блок – *селектор*. Он осуществляет – обращение к БЗ, в которой хранятся соответствующие правила. В ситуационной модели МППР условия описываются непосредственно в элементах модели – преобразователях, агентах.

Для описания событий, которые могут возникать в результате обработки транзактов, изменения состояний объектов и поступления внешней информации используется второй уровень SIE-модели, называемый событийным. Если осуществить проекцию структурного уровня на событийный, то для каждого блока будут существовать, по крайней мере, два события *начало обработки транзакта и окончание обработки транзакта,* два состояния (*занят и не занят*) *и* один процесс (*обработка транзакта*) [18]. В ситуационной модели МППР явного разделения на уровни нет, существует единый процессно-событийный уровень, который также отражает процессный аспект.

Третьим уровнем SIE-модели является ситуационный уровень. Он предназначен для укрупненного моделирования системы. Каждому объекту на структурном уровне сопоставляется микроситуация. Некоторые объекты объединяются в один объект более высокого уровня. Вводятся абстрактные объекты или понятия, для которых определяются возможные ситуации. Ситуации и некоторые микроситуации определяются на основании правил, заложенных в ЭС. Микроситуации и ситуации связаны друг с другом, т. е. они зависят от общих (пересекающихся) атрибутов. Ситуационное моделирование заключается в задании некоторых характеристик, отдельных ситуаций и в определении с помощью ЭС оказываемых влияний. Особняком стоит экспертный уровень, который представляет собой базу знаний, хранящую всю информацию об остальных уровнях и дополнительные знания ЭС [18].

В системах с большой размерностью количество ситуаций и микроситуаций может быть велико. Иногда для упрощения вводят понятия уровня абстракции, т. е. выделяются области системы, для них определяются отдельно макроситуации, и на следующем уровне они

рассматриваются как микроситуации или ситуации [18]. В ситуационной модели МППР уровням абстракции соответствуют уровни системного графа высокого уровня интеграции.

На структурном уровне SIE-модели составляется структурная схема системы, которая представляет собой совокупность связанных блоков. Под блоком понимается сложный объект (модель), в котором связи между внутренними и внешними атрибутами задаются с помощью формализмов. Каждый блок имеет условное графическое обозначение [18].

Совокупности типовых блоков (*Устройство; Канал связи; Поток: Персонал* (*исполнитель работ*)*; Распаковщик; Сборщик; Транзакт; Очередь; Селектор*) структурного и событийного (Событие, Процесс) уровня SIE-модели, ориентированной на предметную область допечатных процессов, в ситуационной модели МППР может быть представлена совокупность ресурсов (*Res*), сообщений (*Message*), заявок (*Order*), средств (*Mech*), преобразователей (*Op, PR, Junction, Resiver, Sender*), очередей заявок (*Queue*). В основе структурного и событийного уровня SIE-модели, реализующего компонент СИМ, используется оригинальная технология виртуальных транзактов. Соответствие математической модели процесса преобразования ресурсов и схемы массового обслуживания было показано в [33, 38].

Экспертный уровень представляет собой базу знаний, в которой хранятся следующие описания:

1) типовых элементов (объектов, блоков) модели;
2) схем различных уровней;
3) программ-селекторов;
4) правил диагностирования ситуаций;
5) дополнительных правил, объектов и связей предметной области;
6) накопленного опыта экспертов и работы системы.

Для хранения информации о типовых элементах используется фреймовый подход. *Схемы уровней*. Каждая схема уровня представляет собой сеть, узлами которой являются типовые блоки.

Программы-селекторы представляют собой набор правил или методов, выполнение которых инициируется обращением к блоку селектора на структурном и событийном уровнях. В простейшем случае селектор реализует условный оператор, который *в*свою очередь представляет собой продукцию вида *Если-То*.

Ситуационный язык. Для описания объектов в БЗ, отношений между объектами и в качестве языка машины логического вывода на

экспертном уровне SIE-модели используется специальный ситуационный язык, базирующийся на логике предикатов первого порядка и фреймах. В общем случае ситуационный язык может быть реализован в виде текстового запроса на ограниченном естественном языке (семиотическом), исчисления предикатов, визуального интерфейса. Разработанный язык позволяет [18]:

– создавать новые и удалять старые объекты СМ;

– изменять характеристики существующих объектов СМ;

– выдавать подробную информацию об объектах;

– осуществлять многокритериальный поиск объектов.

Анализ SIE-модели позволяет сформулировать следующие выводы:

1. SIE-модель, предложенная А.Ю. Филипповичем, может служить базой для создания ситуационной модели МППР.

2. Рассмотренная в данном разделе модель имеет следующие достоинства:

– аппарат/механизм диагностирования ситуаций;

– сочетание имитационного, экспертного и ситуационного подходов.

3. SIE-модель не удовлетворяет следующим требованиям модели МППР:

– наличию модели ЛПР (агента) и сообществ агентов (мультиагентной системы);

– проблемной ориентации на процессы преобразования ресурсов.

Выводы по разделу 2.2

1. Математическая модель процесса преобразования ресурсов, предложенная в [33], была дополнена следующими действиями:

– расширена целями, агентами, ситуациями, макро- и микроситуациями, командами управления;

– усилена аппаратом моделирования конфликтов, возникающих на общих ресурсах и средствах за счет введения модели поведения средства и расширения модели поведения операции;

– интегрирована с аппаратом ситуационного управления в части алгоритма планировщика.

2. Необходимо решить задачи выбора модели интеллектуального агента (модель ЛПР) и интеграции ее с ситуационной моделью процесса преобразования ресурсов.

Следующим шагом является выбор соответствующей математической модели интеллектуального агента (ИА).

2.3. Анализ и выбор модели интеллектуального агента

ИА должны обеспечивать множественные реакции на происходящие в информационном пространстве события и возникающие ситуации, накапливать данные о прошедших событиях и ситуациях, обладать способностью к извлечению знаний и модификации моделей окружающей среды [21, 83].

В [21] приводится следующая классификация архитектур агентных систем [120–121] и соответствующих им моделей интеллектуальных агентов:

– делиберативные архитектуры и модели (deliberative architectures);

– реактивные архитектуры и модели (reactive architectures);

– гибридные архитектуры и модели (hybrid architecture).

В ходе исследования были выделены следующие модели динамического моделирования ситуаций, поддерживающие агентное представление ОТС: модель GAIA, модель Бугайченко Д.Ю., модель Маслобоева А.В., имитационная модель взаимодействия интеллектуальных агентов, модель Ресурсы-Действия-Операции (РДО).

Модель GAIA [146] предложена М. Вулдриджем, Н. Дженнингсом и предназначена для описания системы как искусственной организации, состоящей из разнородных агентов, взаимодействующих друг с другом для достижения глобальной цели. Наиболее абстрактной сущностью в иерархии концептов GAIA является система, следующий уровень иерархии это роли. Ролям ставятся в соответствие типы агентов системы. Функционал агента описывается при помощи модели служб, в которой каждая служба сопоставляется с деятельностью роли. Служба представляет собой преобразователь ресурсов.

Модель Бугайченко Д.Ю. [147] описывает интеллектуальных агентов (ИА), имеющих ментальную (BDI) архитектуру. Свойства МАСавтор модели описывает при помощи разработанной формальной логики MASL – метода логической спецификации мультиагентных систем с временными ограничениями, способных к накоплению и анализу опыта.

Модель Маслобоева А.В. [148] расширяет существующую InteRRap-архитектуру ИА за счет добавления в ее состав проблемно-ориентированной подсистемы непрерывного имитационного моделирования (комплекса системно-динамических моделей), которую агенты используют для имитации сценариев развития ситуаций и прогнозирования последствий своих действий.

Имитационная модель взаимодействия интеллектуальных агентов (ИМВИА) [149] была разработана Рыбиной Г.В. и Паронджановым С.С.

с целью формализации коммуникативной деятельности ИА с помощью моделирования отдельных компонентов коммуникации: состава участников взаимодействия, коммуникативной среды, проблемной области, языка взаимодействия, сценариев диалога. Взаимодействие ИА осуществляется путем диалога, включающего глобальную, тематическую и локальную структуры. Для описания взаимодействия авторами модели используется теория речевых актов (ТРА).

Модель РДО [150] предназначена для описания сложных дискретных систем (СДС) и протекающих в них событий с целью изучения статических и динамических характеристик событий. Система в модели на концептуальном уровне представима в виде множества ресурсов, выполняющих определенные действия. Действия описываются операциями, представляющими собой модифицированные продукционные правила, учитывающие временные связи. Операция с нулевым временным интервалом называется точкой решения и является аналогом реактивного агента, хранящего в базе знаний ответные реакции на воздействие среды.

В работе [21] излагается подход к концептуальному пониманию природы МАС, базирующийся на следующих особенностях:

– необязательно (во многом сомнительно) стремление развивать в технических системах такие понятия, как убеждения, намерения, желания. По мнению автора, здесь имеют место терминологические манипуляции, подменяющие понятия целей, задач и знаний интеллектуальной технической системы, которые, естественно, могут иметь разную функциональную ориентацию;

– технические системы должны быть управляемы и контролируемы человеком, иначе возможны непредсказуемые последствия [123];

– модели мультиагентных интеллектуальных систем, выраженные в терминах специальных логических формализмов, вряд ли могут быть понятны большинству инженеров-программистов, менеджеров информационных проектов и т. д. Даже для инженеров по знаниям применение таких сложных моделей требует специальной подготовки.

Для проведения сравнительного анализа моделей были выбраны следующие критерии сравнения систем: модель преобразователя ресурсов (модель дискретного бизнес-процесса); модель системы массового обслуживания (СМО) – инструмент, хорошо зарекомендовавший при анализе дискретных стохастических систем; модель реактивного агента и модель ИА (инструмент для описания моделей ЛПР). Сравнительный анализ рассмотренных моделей динамического моделирования ситуаций приведен в табл. 2.1.

Таблица 2.1

Сравнительный анализ мультиагентных моделей динамического моделирования ситуаций

Критерии сравнения	Модель Бугайченко	Модель Маслобоева	Модель ИМВИА	Модель GAIA	Модель РДО
Модель преобразователя ресурсов:					
– вход/выход/условия запуска/длительность	НЕТ	НЕТ	НЕТ	+/+/+/ НЕТ	+/+/+/+
– иерархическая модель преобразователя				НЕТ	НЕТ
– вре́менное прерывание операций				НЕТ	НЕТ
Модель системы массового обслуживания	НЕТ	НЕТ	НЕТ	НЕТ	+
Модель реактивного агента (форма представления знаний)	Темпоральная логика MASL	Продукции	Раскрашенные сети Петри	НЕТ	Продукции
Модель интеллектуального агента:					
– наличие у агента целей	+	+	+	+	НЕТ
– прогнозирование действий	+	+	НЕТ	НЕТ	
– анализ и планирование действий	+	+	+	+	
– модель самообучения агентов	+	+	+	НЕТ	
– технология реализации компонент планирования, прогнозирования, обучения	ЭС на основе разрешающих диаграмм	Системно-динамическоеИМ	ЭС	ЭС	
– язык обмена сообщениями	ТРА / Сигналы	ТРА	ТРА	Сигналы	
– модель кооперации агентов	+	+	+	+	
Программная реализация модели	Тестовая реализация	Прототип МАС	Прототип МАС ИМВИА	НЕТ	Прототип РДО-студия

Как следует из табл. 2.1, полный функционал интеллектуального агента реализован в моделях Бугайченко Д.Ю., Маслобоева А.В.. При этом агент у Бугайченко Д.Ю. не отождествляется с ЛПР, а представляет собой программную сущность, работающую независимо от эксперта (аналитика). Модель Маслобоева А.В. использует в качестве средства прогнозирования действий агентов комплекс непрерывных системно-динамических моделей, не поддерживающих описание дискретных процессов, протекающих в исследуемых ОТС. Модель ИМВИА ориентирована прежде всего на исследование способов коммуникаций между агентами и не поддерживает интеграцию моделей агентов с системой ИМ. Модель GAIA объединяет в себе модели преобразователя информационных ресурсов и ИА, однако по реализованному функционалу данная модель уступает РДО.

Модель РДО в области МППР обеспечивает аналитика большим функционалом формализации процессов ОТС. Серьезным недостатком данной модели является отсутствие возможности реализации ИА.

Таким образом, актальной задачей является разработка моделиМППР объединяющего в себе модель гибридного агента (интеллектуального и реактивного), модели преобразователя ресурсов с элементами систем массового обслуживания (СМО). Модели преобразователя и СМО позволят аналитику проводить анализ динамических характеристик ОТС. Модель гибридного агента позволит разрешать задачи анализа в рамках прогона различных сценариев поведения ЛПР. В следующем разделе решается задача выбора модели представления знаний для формализации сценариев ИА.

2.4. Анализ и выбор моделей представления знаний

Одной из важных задач проектирования интеллектуальных систем (ИНС) предприятия является выбор и построение таких моделей представления знаний о предметной области, для которых переход от неформализованных знаний и представлений к формальным моделям и БЗ будет наиболее простым и естественным. Процесс извлечения и приобретения знаний вызывает большие сложности при построении ИНС, ставя перед инженерами по знаниям следующие проблемы [21, 126]:

- неудачный выбор метода извлечения знаний, не соответствующего структуре предметной области;
- неадекватные модели и языки для представления знаний;

• невозможность построения целостной модели предметной области в результате извлечения фрагментов знаний;

• потеря части знаний о предметной области в результате упрощения пространства эксперта;

• зависимость качества создаваемой системы от квалификации «посредников» между экспертами и инструментальными средствами – инженеров по знаниям;

• автономное использование интервью не позволяет найти и устранить «пробелы» в знаниях, интервью субъективно и требует больших затрат времени;

• для приобретения знаний из примеров необходимо обеспечение совместимости БД, имеющих различные схемы, с базой знаний ИНС;

• необходимо преобразование результатов работы алгоритмов обучения на примерах в способ представления, поддерживаемый программными ИНС;

• в тексте отсутствует в эксплицитном виде информация о свойствах элементов текста (имен, предикатов, предложений), необходимая для работы методов приобретения знаний из текстов;

• выполнение семантического анализа текста осложнено отсутствием заранее заготовленного словаря предметной области.

Сложность этапа структурирования знаний проявляется в том [21], что необходимо построить такую модель предметной области, которая позволяла бы наиболее адекватно и с наименьшими усилиями перейти к последующей технической реализации системы. В этом смысле концептуальное моделирование следует рассматривать как процесс порождения, распознавания или нахождения релевантных концептов и концептуальных моделей, описывающих область существования и функционирования информационной системы [127]. *Таким образом, минимизация усилий при переходе от модели предметной области к ее технической реализации является актуальной задачей.*

Следуя H. Kangassalo [128], концепт рассматривается как фундаментальное понятие, описывающее именованный независимо идентифицируемый структурированный конструкт, объединяющий примитивы знаний. Концепт является интенсиональной структурой знаний, которая содержит неявные правила, ограничивающие структуру реальности, а также концентрирует и организует информацию, необходимую для структурирования и понимания основных аспектов знания и характеризует некоторые свойства обозначаемых им объектов [21].

2.4.1. Анализ фреймовых моделей

В концептуальной модели предметной области необходимо соединить описание структуры предметной области, определить поведение объектов и субъектов, существующих в этой структуре, построить логические модели их взаимодействия [21]. Minsky в своей работе [29] определил **фрейм** как «**структуру данных для представления стереотипных (стандартных) ситуаций**». Эту структуру он наполнил самой разнообразной информацией: об объектах и событиях, которые следует ожидать в этой ситуации, и о том, как использовать информацию, имеющуюся во фрейме. Идея состояла в том, чтобы сконцентрировать все знания о данном классе объектов или событий в единой структуре данных, а не распределять их между множеством более мелких структур вроде логических формул и порождающих правил. Такие знания либо сосредоточены в самой структуре данных, либо доступны из этой структуры (например, хранятся в другой структуре, связанной с фреймом) [119].

С каждым фреймом ассоциируется разнообразная информация (в том числе и процедуры); например, информация о том, как пользоваться данным фреймом, каковы ожидаемые результаты выполнения фрейма, что делать, если ожидания не оправдались, и т. п. Фрейм можно представить в виде сети, состоящей из вершин и отношений (дуг) [11].

Преимущества подхода, основанного на фреймах, заключаются в следующем [21]: концепция фреймов естественным образом интегрируется с концептуальным моделированием предметной области; структуры фреймов хорошо описываются средствами объектно-ориентированного проектирования; эффективно поддерживаются возможности наследования; обеспечивается иерархическое представление предметной области. *Таким образом, выбор фреймов обосновывает применение объектно-ориентированного подхода и объектных языков программирования при разработке СДМС и минимизирует затраты на создание программного обеспечения.*

Анализ работ, выполненный А.Н. Швецовым в [21], по объектно-ориентированному проектированию и программированию позволяет выделить три основных класса моделей, представляющих объекты и классы (англ. class-based representation formalism): модели, основанные на семантических сетях и фреймах; модели, развиваемые на основе теории баз данных и семантических моделей данных; модели, использующие работы по абстрактным типам данных. *Как показало*

исследование, основанные на фреймах языки используются для расширения дедуктивных возможностей семантических и объектно-ориентированных моделей данных, что и обосновывает их использование и проведение дальнейших исследовательских работ в данном направлении.

2.4.2. Применение фреймового подхода Швецова для построения концептуальной модели предметной области (КМПО)

Для построения концептуальной модели предметной области и решения задачи сокращения затрат на разработку ПО, в работе использован фреймовый подход Швецова [21], основанный на совмещении фреймоподобных структур с конструкциями концептуальных графов J.F. Sowa [129–131]. Преимуществами данного подхода является деление на активные и пассивные фреймы и учет поведения объекта.

Основная конструкция фрейм-концепта (ФК) представлена в кортежной модели 2.33. Имя фрейма представляет собой уникальный идентификатор, используемый в КМПО. Информация о применении на уровне ФК является неформальным вербальным описанием возможных ситуаций использования ФК, сценариев поведения, особенностей выбора и т. п. Динамическое поведение компонентов или агентов предметной области описывает структура сценариев поведения (ССП), в которую включен блок выбора сценария (БВСЦ), позволяющий формировать альтернативные пути поведения данного фрейма [21].

Структура слотов (ССЛ) представляет собой совокупность двух структур: структуры концептов (СК) и структуры атрибутов (СА). СК содержит список фрейм-концептов, в некотором отношении вложенных или порожденных охватывающим ФК, тип этого отношения указывается в поле «тип концептуального отношения», т. е. отношение данного $ИК_i$ к ФК, где $ИК_i$ – имя *i*-го концепта. Для установления логической организации предметной области ФК соединяются в структуры концептуальных графов. Концептуальный граф (КГ) есть двудольный граф, имеющий два типа вершин: вершины концептов, или концептуальные вершины, и вершины концептуальных отношений (КО) [21]. Таким образом, Швецовым предлагается использовать фреймово-семантическое представление знаний.

Модель фрейм-концепта определяется следующим образом [21]:

$$ФК = <ИФ, ТФ, ИП, ССП, ССЛ>, \quad (2.33)$$

$$ССЛ = <СК, СА> , \quad (2.34)$$

$$СК = \{(ИК_1, КО_1), (ИК_2, КО_2), ..., (ИК_n, КО_n)\}, \quad (2.35)$$

$$СА = \{(ИА_1, МО_1, ЗА_1), (ИА_2, МО_2, ЗА_2), ...,(ИА_m, МО_m, ЗА_m)\}. \quad (2.36)$$

где ИФ – имя фрейма;
ТФ – тип фрейма;
ИП – информация о применении;
ССП – структура сценария поведения;
ССЛ – структура слотов;
СК – структуры концептов;
СА – структуры атрибутов;
$ИК_n$ – имя концепта;
$КО_n$ – концептуальное отношение;
$ИА_m$ – имя атрибута;
$МО_m$ – множество определения;
$ЗА_m$ – значение атрибута.

Таким образом, применение подхода к описанию предметной области в виде ФК и КГ, предложенное Швецовым, позволяет использовать фреймово-семантическую модель представления знаний. Задача перехода (совмещения) модели представления знаний, концептуальной модели и их технической реализации на уровне базы данных решается в разделе 3.3.2.

2.5. Алгоритмы работы системы (машина вывода)

Выбор метода поиска решений и реализации механизмов вывода определяется целым рядом факторов [21]: спецификой предметной области, размерами пространства поиска, уровнем определенности и надежности знаний и данных, динамикой происходящих в предметной области изменений, моделями представления знаний, вычислительными ресурсами, которые могут быть применены в интеллектуальной системе.

Задача построения алгоритма ситуационного моделирования разбивается на две: создание алгоритма работы ИА (раздел 2.5.1) и алгоритма планировщика работы ситуационной модели (раздел 2.5.2), на которого возлагаются задачи обработки действий элементов процесса преобразования ресурсов и множества ИА.

2.5.1. Алгоритм работы интеллектуального агента

Функции анализа, структурирования, обобщения и экстраполяции ситуаций, а также выработку команд управления процессом преобразования ресурсов целесообразно возложить на агентов. Причем для каждого агента необходимо обеспечить доступ к его базе знаний.

Для описания работы агента введем следующие массивы и операторы:

Mess_Buffer – массив «Буфер сообщений». Для каждого агента в данном массиве резервируется переменная (*Mess_In_Count*), отражающая количество входящих сообщений. В случае если данная переменная не равна нулю, то происходит вызов метода агента *Agent.R_Mess.* Возвращение значения *Mess_In_Count* происходит путем вызова функции *Mess_Buffer.Check_Mess*(*Agent.name*). Запись сообщений агентом в буфер происходит вызовом метода *Agent.W_Mess*. После вызова методов *R_Mess* и *W_Mess* значения переменных *Mess_In_Count* и *Mess_Out_Count*, соответственно, сбрасываются в ноль.

За основу правил поведения агентов взят специализированный объектно-ориентированный язык *RADL* (Reticular Agent Definition Language) в виде конструкции **When-If-Then**, реализованный в системе проектирования агентов и мультиагентных систем *Agent Builder* (Reticular Systems, Inc.) [92]. Ментальная модель включает описание намерений, желаний, обязательств и возможностей, а также правил поведения агентов. На основе этой модели осуществляется выбор тех или иных действий интеллектуального агента. Составные части этого правила выполняют следующие функции: **When** <...> содержит новые сообщения, полученные от других агентов; **If** <...> сравнивает текущую ментальную модель с условиями применимости правила; **Then** <...> определяет действия, соответствующие текущимсобытиям, состоянию ментальной модели и внешнего окружения.

Применительно к предметной области процессов преобразования ресурсов, с учетом того, что ментальную модель агента

представляет модель целеполагания, функция **When** будет непосредственно включена в алгоритм, то будем использовать следующую структуру правил поведения агентов:

Name <*Имя правила*>

If <*Message Conditions, RCP Conditions, G_Ag Conditions*>

Then <*G_Ag Changes; Message Actions; Private Actions*>.

где *MessageConditions* – условия, относящиеся к сообщениям;
RCPConditions – условия, относящиеся к преобразователям и ресурсам (*RCP – Resources conversion process*);
G_AgConditions – *условия*, относящиеся к целям;
G_AgChanges – действия по изменению текущих целей агента;
MessageActions – действия по формированию сообщений;
PrivateActions – действия (план действий), связанные с преобразователями и ресурсами, направленные на достижение поставленных целей.

Части правил могут быть представлены в виде предикатов первого порядка. Следуя [21, 133–134], полагаем, что предикат от *n*еременных, *n*-арный предикат на множестве *A* есть *n* – местная функция, определенная на *A* со значениями в множестве {истина, ложь}. Совокупность наборов (a_1, a_2, ..., a_n) элементов из *A,* для которых $P(a_1 a_2, ..., a_n)$ – истина, называется *n*-арным отношением, отвечающим предикату *P.* И наоборот, любому *n*-арному отношению *R* на *A* отвечает предикат $P(x_1, x_2, ..., x_n)$

$$P(a_1, a_2, ..., a_n) = \begin{cases} \text{истина, если } (a_1, a_2, ..., a_n) \in R, \\ \text{ложь, если } (a_1, a_2, ..., a_n) \notin R. \end{cases}$$

Часть ***Then*** включает в себя действия и планы действий, т. е. набор последовательно-параллельных действий с привязкой ко времени или решение ***Decision***.

Алгоритм работы агента состоит из следующих действий:

1. **Анализ мира (окружения) агента** включает:
   i. Чтение сообщений «Агента-начальника» и «Агентов-подчиненных» (функция **When**), если таковые есть.

$$Agent.Mess\_In\_Count := Mess\_Buffer.Check\_Mess(Agent.name)$$

$$IF\ \ Agent.Mess\_In\_Count > 0$$

$$THEN\ \ Agent.R\_Mess$$

В случае если при чтении сообщения определяется конфликтная ситуация – входящее сообщение противоречит текущим целям агента, то происходит разрешение конфликта целей (более приоритетные цели вытесняют менее приоритетные). Если агент не принимает к выполнению входящее сообщение, то формируется отказ.

ii. Обзор (анализ) ближайшего окружения, элементов процесса преобразования, которыми он управляет непосредственно.

iii. Контроль выполнения целей.

2. **Диагностирование ситуаций и Поиск решения.** Диагностирование ситуации заключается в сопоставлении части IF правил с текущим состоянием мира агента. В зависимости от найденной (диагностируемой) ситуации агент обращается к БЗ и пытается найти соответствующее решение, которое переведет систему из текущей ситуации в желаемую. В случае если диагностировано более одной ситуации, то выполняется анализ их непротиворечивости, ранжирование по приоритетам и значимости для агента, формирование общего решения («склейка» команд, устранение повторов).

В случае если найдено несколько решений, то для каждого решения агент проводит оценку его достижимости (моделирует поведение всех элементов, принадлежащих данному модельному множеству для поиска пути к цели ga^j на глубину в gp шагов). Полагаем, что параметр gpопределяется выбранной стратегией s^i, и вводим два счетчика: счетчик модельных шагов $Cmgp$и счетчик актуальных шагов $Cagp$. Выполняем модельные шаги для модельных множеств на глубину gp. После каждого шага вычисляем предполагаемую оценку состояния мира ИА для всех моделируемых информационных объектов (ИО – элементов процесса преобразования) и делаем следующий модельный шаг. Повторяем модельные шаги до достижения $Cmgp = gp$. Затем сравниваем оценки достижения цели текущего решения с предыдущим, запоминаем решение с лучшей оценкой. Далее вычисляем предполагаемую оценку преобразований целей для всех ИО и отклонение от целей (меру близости к цели):

$$\Delta_{\min} = \min\left(\Delta\left(IO_m^{j,i}\right) \mid IO_m^{j,i} \in ModelSet^j\right).$$

Таким образом, перебрав все решения, выбираем единственное.

Решению должен соответствовать набор действий как самого агента, так и элементов процесса преобразования ресурсов,

подчиненных ему и обеспечивающих переход системы в желаемое состояние.

$$SR(Agent_i): GS^{Agent_i} \rightarrow VO^{Agent_i},$$
$$VO^{Agent_i} = \{Op^{Agent_i}\} \cup \{PR^{Agent_i}\} \cup \{Sender^{Agent_i}\} \cup$$
$$\cup \{Receiver^{Agent_i}\} \cup \{Junction^{Agent_i}\} \cup \{Agent^{Agent_i}\}.$$

Если ситуация не изменилась, то переходим к следующему шагу.

Генерация (выработка) **решения** (если решения в БЗ не найдено). Если ситуация диагностирована и нет ни одного решения, значит пользователь (эксперт) либо не знает решения, либо не ввел его. В первом случае развитие системы идет в соответствии с ранее заданной стратегией (набором управляющих воздействий), во втором случае пользователь вводит решение (пополняет БЗ).

3. **Корректировка целей**. Агент, диагностировав ситуацию и выбрав решение, корректирует свои цели и цели вверенных ему объектов. Вычисляет оценки состояния целей от достигнутого состояния мира ИА и определяет отклонения от цели. Поскольку целей может быть несколько (в случае $j > 1$), необходимо сформировать комплексную оценку положения ИА в пространстве целей

$$\Delta_k = \sum_{j=1}^{n} a_i \Delta\left(ga^j, g\hat{a}^j\right),$$

учитывающую значимость целей, a_i – весовые коэффициенты важности целей, n – мощность множества активных целей.

Проводим сравнение с допустимым комплексным отклонением Δd с помощью следующих правил:

при выполнении условия $\Delta_k > \Delta d$ переходим к шагу (2), изменяя при этом стратегию, т.е. переоцениваем стратегию формирования целей; если же $\Delta_k \le \Delta d$, то продолжается функционирование ИА в той же стратегии.

4. **Формирование модели мира** данного агента (прогнозирование состояния на следующем шаге) $FV(t + 1)$.Формируем новое текущее состояние мира ИА с помощью вычисления значений целей. FV – функция формирования информационного пространства.

5. Выполнение активных действий агента происходит на текущем шаге модельного времени

$$Execute: AD(Agent_i): GA \rightarrow VO,$$

где *Execute* – функция выполнения действий агента. К активным действиям также относится функция отправки сообщений

IF Agent.Mess_Out_Count > 0
THEN Agent.W_Mess

6. Окончание выполнения действий агента на данном шаге.

На рис. 2.6 приведена схема работы ИА.

2.5.2. Алгоритм ситуационно-имитационного моделирования

С целью минимизации вычислений, выполняемых машиной вывода, предлагается использовать алгоритм, в котором учет модельного времени организован с использованием централизованного календаря событий. За основу взят алгоритм, предложенный в [33–34] и доказавший свою эффективность и быстродействие при решении практических задач в системе BPsim. В процессе работы машина вывода *Ips* взаимодействует со следующими очередями:

Календарь (*Calendar*) – очередь, содержащая упорядоченный по модельному времени перечень правил, которые должны выполнить одно из действий в определенный момент времени (проверить *условие запуска* $C_a(t)$, выполнить действие $A_{IN}(t)$, действие $A_{OUT}(t)$) или действие агента $A_{Agent}(t)$:

$$Calendar = A_1(t_1),\ A_2(t_2), ..., A_j(t_j), ...A_m(t_m);$$

$$\forall j : \left(A_j(t) \in \left\{ A_{Ca}(t) \right\} \cup \{A_{IN}(t)\} \cup \left\{ A_{OUT}(t) \right\} \cup \left\{ A_{Agent} \right\} \right) \wedge \left(t_{j-1} \le t_j \le t_{j+1} \right). \quad (2.37)$$

Список активизированных правил (*Active_Rules*) представляет собой перечень правил, которые должны в данный момент времени выполнить действие $A_{OUT}(t)$:

$$Active\_Rules = A_1(t_1), ..., A_i(t_i), ..., A_n(t_n);$$

$$\forall i : \left(t_i = SysTime \right) \& \left(A_i(t_i) \in \left\{ A_{OUT}(t) \right\} \right), \quad (2.38)$$

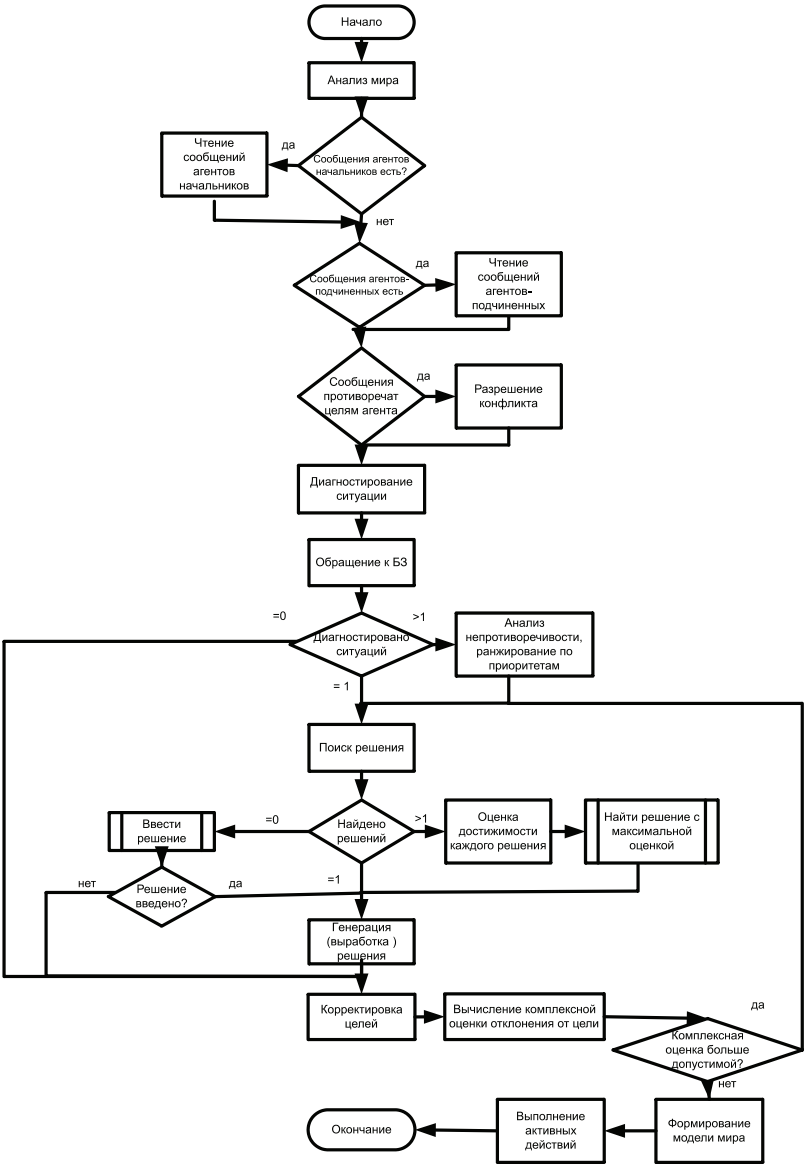


Рис. 2.6. Блок-схема работы ИА

Очередь правил (*Wait&Lock_Rules*), состоящая из «операций», «перекрестков», «источников» и «приемников» – правил, у которых будет выполняться в данный момент действие $A_{Ca}(t)$, для правил с признаком прерывания $C_a^{Mech}(t)$ и для действий агентов $A_{Agent}(t)$:

$$Wait\ \&\ Lock\_Rules = A_{11}(t\ ),...,A_k(t_k),...,A_z(t_z);$$

$$\forall k:\left(t_k = SysTime\right)\ \&\left(A_k(t_k)\in\left\{A_{Ca}(t)\right\}\right). \tag{2.39}$$

Для реализации методов ситуационного управления дополнительно в алгоритм введены следующие очереди:

Список агентов (*Agent_list*), перечень всех агентов, упорядоченный по приоритету. Агенты с наивысшим приоритетом обрабатываются в первую очередь. Приоритет изменяется от 0 до *M*, причем 0 – наивысший приоритет.

$$\begin{gathered} Agent\_List = AG_1, AG_2, ..., AG_p, ..., AG_y \\ \forall p: prior_{p-1} \geq prior_p \geq prior_{p+1}, \end{gathered} \tag{2.40}$$

Для работы с очередями введем следующие операторы:

$A_j \xrightarrow{ADD} Rules\_Queue_S$ – добавление в *S*-ю очередь *j*-го действия правила;

$Rules\_Queue_R \xrightarrow{DEL} A_i$ – удаление из *R*-й очереди *i*-го действия правила которое было выполнено;

$Rules\_Queue_Q \xrightarrow[A_O]{REMOVE} Rules\_Queue_W$ – перемещение *o*-го действия правила из *Q*-й очереди в *W*-ю;

Clear ($Rules\_Queue_X$) – очистка *X*-й очереди от элементов;

Count ($Rules\_Queue_C$) – функция, которая определяет количество элементов в очереди.

Для запоминания изменений рабочей памяти на текущем шаге введем очередь *Changes_Queue*, которая хранит перечень элементов *Rps,* изменение состояния которых произошло в текущем проходе машины вывода.

Execution ($Rules\_Queue_W$) – функция выполнения действий (изменение состояний ресурсов, механизмов, заявок и сообщений) правил *W*-й очереди. Последовательность обработки действий определяется приоритетом правил.

Машина вывода выполняет в момент запуска подготовительный этап – устанавливает значение модельного времени *SysTime,* равное нулю или моменту, задаваемому пользователем. В дальнейшем машина вывода циклически выполняет следующую последовательность шагов:

1. **Определение текущего момента времени** – определяется ближайший момент завершения активности по календарю:

$$SysTime = \min_{j \in A^{Calendar}} T_j.$$

В случае моделирования непрерывных процессов преобразования ресурсов $SysTime = SysTime + \Delta t$, т.е. имеем пошаговый способ продвижения по модельному времени.

2. **Определение списка активизированных событий** для момента времени *SysTime*, в том числе:

1) список завершения активности экземпляров «перекресток» с заданным временем синхронизации:

$$\forall i \left(t^{out}_{Junction_i} = SysTime \right):$$
$$A^{Junction_i}_{OUT} \xrightarrow{ADD} Active\_Rules;$$

2) список завершения активности экземпляров «операция»:

$$\forall k \left(Status^k_{Op} = "active" \right) \& \left(t^{out}_{Op_k} = SysTime \right):$$
$$A^{Op_k}_{OUT} \xrightarrow{ADD} Active\_Rules;$$

3) список завершения активности экземпляров «источник» и «приемник»:

$$\forall m \left(t^{out}_{receiver_m} = SysTime \right): A^{receiver_m}_{OUT} \xrightarrow{ADD} Active\_Rules;$$
$$\forall n \left(t^{out}_{sender_n} = SysTime \right): A^{sender_n}_{OUT} \xrightarrow{ADD} Active\_Rules.$$

3. **Обработка списка агентов**. Если в модели присутствует хоть один агент, то Планировщик последовательно, согласно приоритетам, совершает обход агентов (с передачей им полномочий), работает со списком *Agent_List*. При выполнении 5-го шага алгоритма агента (см. раздел 2.5.1) машина вывода помещает действия агента в очередь правил или в календарь (если действие не будет выполняться

в текущем такте). Список выполнения действий экземпляров «агент», запланированных на данный момент системного времени, следующий:

$$\forall d \left(t_{Agent_d} = SysTime \right) : A^{d}_{Agent} \xrightarrow{ADD} Active\_Rules.$$

4. **Исполнение действий** (соответствующих активизированным правилам и определенным на предыдущем шаге) и фиксация изменений в рабочей памяти. Завершение активизированных правил *Execute* (*Active_Rules*). Для экземпляров «источник», «приемник», «операция», «агент», условием запуска которых является функция от времени – определение (вычисление) следующего момента F_x^{time} активизации и записи его в календарь:

$$Active\_Rules \xrightarrow[A_X]{REMOVE} Calendar.$$

Для экземпляров, у которых условие запуска содержит помимо временной составляющей иные условия, выполняется проверка условия запуска. В случае невыполнения условия запуска происходит перемещение из календаря в очередь правил:

$$Calendar \xrightarrow[A_X]{REMOVE} Wait\ \&\ Lock\_Rules.$$

Экземпляры, которые были завершены в данном такте, помещаются в очередь правил с целью дальнейшего анализа (экземпляры без условий).

5. **Проверка рабочей памяти.** Если состояние памяти изменилось *Count* (*Changes_Queue*) ≥ 0, то происходит переход к следующему шагу, иначе – переход на 10-й шаг.

6. **Формирование очереди правил.**

Если такт не первый, то машина вывода из списка изменившихся ресурсов и средств формирует очередь правила (используя *Rules_of_use*).

Если такт первый, то машина вывода выбирает все правила. Очередь формируется в следующей последовательности:

1) правила экземпляров «агент»:

$$\forall c : A^{c}_{Agent} \xrightarrow{ADD} Wait\ \&\ Lock\_Rules;$$

2) правила экземпляров «перекресток»:

$$\forall i : A^{Junction_i}_{Ca} \xrightarrow{ADD} Wait\ \&\ Lock\_Rules;$$

3) правила не активизированных и не попавших в календарь экземпляров «источник» и «приемник»:

$$\forall l : A_{Ca}^{receiver_l} \xrightarrow{ADD} Wait \,\&\, Lock\_Rules;$$
$$\forall k : A_{Ca}^{sender_k} \xrightarrow{ADD} Wait \,\&\, Lock\_Rules;$$

4) пассивные правила (т.е. события, которые активизируются не в определенный момент времени, а при выполнении определенного условия) экземпляров «процесс» и «операция» и правила, находящиеся в очереди прерываний.

При формировании очереди внутри каждого типа, ***правила*** выстраиваются в соответствии с приоритетом.

7. **«Обнуление» изменений области рабочей памяти** (снятие признаков изменения состояния ресурсов и средств): *Clear* (*Changes_Queue*).

8. ***О*бработка очереди правил и фиксация изменений в рабочей памяти** *Execute* (*Wait&Lock_Rules*)*:*

1) проверка исполнения правил экземпляров «агент», в случае выполнения правила – запуск активности «агента»;

2) проверка исполнения правил экземпляров «перекресток», в случае выполнения правила – запуск активности «перекрестка» (блокировка или удаление входных сообщений и формирование выходных);

3) проверка исполнения пассивных правил и тех, которые не попали в календарь экземпляров «источник» и «приемник», в случае выполнения правила – запуск активности (блокировка или удаление входных сообщений и формирование выходных);

4) проверка исполнения пассивных правил экземпляров «процесс» и «операция» и правил с признаком прерывания (с последующей блокировкой механизмов). В случае выполнения правила экземпляр будет активизирован (захват входных ресурсов и средств, расчет времени окончания работы) и правило «удаляется» из очереди.

Данные правила обрабатываются машиной вывода последовательно, начиная с правил высшего приоритета, в следующем цикле:

8.1. Текущее правило. Начало обработки правила. Если текущее правило имеет *признак прерывания*, то переходим *к* следующему шагу, если правило не имеет *признака прерывания*, то шаг 8.3, правил больше нет – шаг 2.

8.2. Условие запуска правила на входных ресурсах выполняется? Если да, то переходим к следующему шагу, иначе – переходим к шагу 8.8.

8.3. Условие запуска на средствах выполняется? Если да, то переходим к шагу 8.5, иначе – переходим к шагу 8.4.

8.4. Правило с абсолютным приоритетом (АП)? Если правило с АП, то переходим на шаг 8.6, иначе – шаг 8.8.

8.5. Обработка правила и фиксация изменений в рабочей памяти (РП). Переход к шагу 8.8.

8.6. Есть ли правила с меньшим приоритетом, использующие недостающие средства? Выбор активизированных правил с меньшим приоритетом и не имеющих *признака прерывания*, использующих необходимые средства в нужном количестве. Если такие правила есть, то происходит переход к шагу 8.7, иначе – переходим к шагу 8.8.

8.7. Выполнение прерывания, обработка правила АП и фиксация изменений в РП. Происходит прерывание правил, определенных на предыдущем этапе (установление *признака прерывания*). Выполнение правила с абсолютным приоритетом и фиксирование изменений в рабочей памяти.

8.8. Следующее правило. Если есть необработанные правила, то осуществляется переход к обработке текущего правила – к шагу 8.1, иначе – переходим к шагу 2.

Фиксация изменений рабочей памяти происходит вместе с активизацией события и записью момента окончания в календарь.

9. **Изменился ли текущий момент?** Определение разницы между ближайшим и текущим моментом времени в календаре. Если разница больше нуля, то происходит переход на следующий шаг (шаг 10), иначе – переход на шаг 2 (для завершения «нулевых» работ).

10. **Проверка условия окончания имитации.** В зависимости от режима остановки имитации:

- окончание времени;
- достижение определенного значения параметра;
- «Если делать НЕЧЕГО».

Если условие окончания имитации выполнено, то моделирование завершено, иначе – машина вывода переходит на следующий цикл (шаг 1).

Выводы

1. Модель процесса преобразования ресурсов [33], была дополнена следующими элементами: целями, агентами, ситуациями, макро- и микроситуациями, командами управления.

2. Исследована возможность построения ситуационной модели МППР на базе семиотической и SIE-модели.

3. Для описания модели предметной области используется фреймово-семантическое представление знаний. Решена задача перехода (совмещения) модели представления знаний, концептуальной модели и их технической реализации на уровне реляционной базы данных (РБД). Такой подход позволил в качестве языка вывода на фреймовой модели использовать Transact SQL.

4. Разработана ситуационная модель МППР на основе интеграции аппаратов ситуационного управления, процессов преобразования ресурсов, фреймовых ЭС, ИМ и мультиагентных систем. В рамках модели МППР решены следующие задачи: определены состав и структура мультиагентной системы преобразования ресурсов; доработаны типы правил преобразования и их структура, усиливающие аппарат моделирования конфликтов, возникающих на общих ресурсах и средствах; для модели МППР разработан алгоритм машины вывода-имитатора, реализующий ситуационный подход.

3. ОПИСАНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ СИТУАЦИЙ BPSIM2

В главе представлены результаты разработки СДМС, позволяющей разрабатывать модели мультиагентных систем и имитировать действия агентов со знаниями. СДМС процессов преобразования ресурсов BPsim2 [141, 143–144] была разработана с использованием средств Borland Delphi 7.0 и СУБД MS SQL Server. Подробное описание работы с системой приведено в [135].

3.1. Функции системы

Система обеспечивает выполнение следующих функций:

1. Проектирование концептуальной модели предметной области.
2. Проектирование и генерирование структуры базы данных предметной области.
3. Создание (копирование / удаление) иерархической модели процесса преобразования ресурсов.
4. Добавление (удаление / корректировка) операции. В рамках операции:
   a) добавление (удаление / корректировка) входного ресурса;
   b) добавление (удаление / корректировка) выходного ресурса;
   c) добавление (удаление / корректировка) средства;
5. Добавление (удаление / корректировка) ресурса.
6. Добавление (удаление / корректировка) средства.
7. Добавление (удаление / корректировка) параметра.
8. Добавление (удаление / корректировка) сообщения.
9. Добавление (удаление / корректировка) заявки.
10. Добавление (удаление / корректировка) источника ресурсов.
11. Добавление (удаление / корректировка) приемника ресурсов.
12. Добавление (удаление / корректировка) перекрестка.
13. Добавление (удаление / корректировка) связи.
14. Добавление (удаление / корректировка) цели.
15. Добавление (удаление / корректировка) интеллектуального агента.
16. Функционально-стоимостный анализ.
17. Проведение имитационных экспериментов.
18. Ситуационное моделирование процессов преобразования ресурсов.

19. Анализ результатов экспериментов.

20. Получение отчетов по структуре процессов преобразования ресурсов и проведенным экспериментам:

a) печать экранных форм описания процесса, операции, приемника, источника ресурсов и перекрестка;

b) текстовый отчет по структуре процесса преобразования ресурсов;

c) текстовый отчет с результатами имитационного эксперимента.

21. Экспорт результатов экспериментов в MS Excel.

22. Экспорт результатов экспериментов в MS Project.

3.2. Принципы построения СДМС процессов преобразования ресурсов

3.2.1. Применение объектно-ориентированного подхода при создании системы

Одним из методов разработки интеллектуальных программных систем, к классу которых относятся СДМС, является объектно-ориентированный подход (парадигма). Этот подход является развитием фреймового представления. В его основе лежат понятия *объект и класс* [11]. Достоинства объектно-ориентированного проектирования заключаются в следующем [21]:

– обеспечиваются основные свойства объектной ориентации – инкапсуляция и наследование;

– поддерживается модульная структура, содержащая множество автономных сущностей, взаимодействующих через обмен сообщениями;

– используется независимость реализации, т. е. внутренний механизм функционирования объекта может модифицироваться без влияния на остальные части системы.

При реализации программной составляющей СДМС использован объектно-ориентированный подход. Как уже было отмечено ранее, использование фреймового представления знаний и ООП позволяет однозначно сопоставить программным объектам реальные объекты предметной области.

3.2.2. Математический аппарат

Математический аппарат используется для описания преобразователей (функций условия запуска, входа, выхода, обработки сообщений), целей, параметров, агентов.

Одним из основных требований к языку BPsim2 является требование простоты. Оно достигается за счет ограниченного числа введенных синтаксических конструкций, представленных в графическом виде. BPsim2 является языком структурного типа, что по общепринятому на данный момент мнению также облегчает разработку имитационных программ [3].

Синтаксические правила задаются с помощью графико-синтаксических диаграмм (интерфейсов), ориентированных на проблемную область процессов преобразования ресурсов.

Перечень операндов МАС преобразователей

В качестве операндов используются следующие множества: ресурсов *RES*; средств *MECH*; заявок *Order*; сигналов *Sig*; целей *G*; агентов *Agent*; сообщений *Message*; параметров *P*. Операторами задания операндов являются следующие: описания ресурса; описания средства; описания заявки; описания цели; описания агента; описания сообщения; описания параметра.

Операторы определения структуры процессов преобразования ресурсов. Для описания модели процессов преобразования ресурсов используются следующие операторы:

- декомпозиции (детализирует сложный процесс преобразования ресурсов на композицию более простых);
- внутренний оператор сопряжения элементов подмодели (в графическом виде задает причинно-следственные связи между элементами подмодели). Связь представляет вектор операндов, формируемый *i*-м элементом на выходе и потребляемый на входе *j*-м элементом подмодели;
- сопряжения подмоделей (задает в графическом виде причинно-следственные связи между элементами модели верхнего $(k-1)$-го уровня декомпозиции процесса преобразования ресурсов и элементами подмоделей (k)-го уровня декомпозиции).

Операторы продукций процесса преобразования ресурсов следующие: задания условия запуска; задания входа; задания выхода; задания средства; задания длительности / интенсивности; задания приоритета.

Операторы продукций интеллектуального агента: задания условия ситуации; задания действия ситуации; задания целей; задания приоритета.

Для описания модели целей сложной системы используются следующие операторы: задания функции вычисления текущего значения цели; определения зоны значений цели.

Перечень математических операций, описывающих условия и действия продукций

В выражениях над операндами в BPsim2 используются следующие стандартные операции:

арифметические операции: «–» – унарный минус; «–» вычитание; «+» сложение; «*» – умножение; «/» – деление; «^» – возведение в степень;

логические операции: «?» – отрицание; «&» – конъюнкция; «|» – дизъюнкция;

операции отношения: «=» – равно; «><» – не равно; «>» – больше; «>=» – больше или равно; «<» – меньше; «<=» – меньше или равно;

присваивания «:=»;

кванторы: «-]» – существования; «\/» – всеобщности; «[-» – принадлежности; «~[-» – непринадлежности;

уточнения «#» (определение значения операнда пользователем во время эксперимента).

Дополнительные операции: генерация случайного значения «randomize()»; экспонента «exp»; логарифм натуральный «ln»; корень квадратный «sqrt»; целая часть от деления «!»; дробная часть от деления «$».

В выражениях могут использоваться скобки «(», «)», «[», «]».

Учитывая специфику процессов преобразования ресурсов и требования к простоте языка, операторные выражения в зависимости от принадлежности операндов к входам/выходам/механизмам/агентам при выполнении различных действий продукции ($Action_{Ca}$, $Action_{IN}$, $Action_{Lock}$, $Action_{UnLock}$, $Action_{OUT}$, $Action_{Mech}$, $Action_{Agent}$) интерпретируются определенным образом. Зависимость интерпретации операторного выражения от действий приведена в табл. 3.1, где F – операторное выражение, определенное пользователем.

Функции для работы с очередями заявок:

1. *SelectAllMessage*([*usl*],[*mUsl*],[*mDo*]) – функция выбора множества заявок, которую следует читать так: пока выполняется общее условие *usl,* выбрать (пометить) все сообщения, для которых выполняется условие *mUsl*; действие *mDo* выполнить столько раз сколько выбрано заявок. Например,

SelectAllMessage

([(*Z6_count* < 300)],[(*Z2_owner* = *@bA0_1*)&(*Z2_count* < 300)], [*Z6_count:* = *Z6_count* + *Z2_count*]), что означает выполнение последовательности действий:

1) условие (Z6_count < 300) выполняется? – Да – п. 2, нет – выход;

2) существует заявка [(*Z2_owner* = *@bA0_1*)&(*Z2_count* < 300)]? Да – п. 3, нет – выход;

3) изменить значение параметра Z6_count;

4) переход на п. 1.

Таблица 3.1

Интерпретация операторных выражений

Действие	Выполняемая операция	Описание
Проверка условия запуска на входном ресурсе *RESi*	(*RESi* – *F*) > = 0, где *RESi* – текущее количество *i*-го ресурса, *F* – количество *i*-го ресурса необходимое для выполнения правила	Если результат вычисления отрицательный, то условие запуска на входном ресурсе не выполнилось.
Проверка условия запуска на механизме *MECHi*	(*MECHi* – *F*) > = 0, где *MECHi* – текущее количество *i*-го механизма, *F* – количество *i*-го механизма, необходимое для выполнения правила	Если результат вычисления отрицательный, то условие запуска на механизме не выполнилось.
Захват входного ресурса *RESi*	*RESi* = *RESi* – *F*	Уменьшение количества ресурса на величину, заданную пользователем
Блокировка механизма *MECHi*	*MECHi* = *MECHi* – *F*	Уменьшение количества ресурса механизма на величину, заданную пользователем
Освобождение механизма *MECHi*	*MECHi* = *MECHi* + *F*	Увеличение количества ресурса механизма на величину, заданную пользователем
Формирование выходного ресурса *RESi*	*RESi* = *RESi* + *F*	Увеличение количества ресурса на величину, заданную пользователем

2. DeleteAllMessage([mUsl],[mDo]) – функция освобождения множества заявок. Для заявок, на которых выполняется условие *mUsl*, выполнить действие *mDo*.

DeleteAllMessage([*Z2_lock* = *@bA0_5_1_1*],[*Z2_owner:* = *@bA0_5_1*])

3. CreateOrder(nam) – создать экземпляр заявки Nam.

4. DeleteOrder(nam) – удалить экземпляр заявки Nam, заблокированный текущим блоком.

5. Count(queue, [p1...pn], [v1...vn]) – функция возвращает количество элементов очереди, удовлетворяющей условию ограничения по параметрам очереди.

6. Select(queue, [p1...pn], [v1...vn]) – функция возвращает номер элемента очереди, удовлетворяющей условию ограничения по параметрам и приоритетам.

7. NewElementQueue(p1...pn) – добавляет новый элемент в очередь и устанавливает его атрибуты.

8. LockCurent(queue) – блокирование текущего элемента очереди.

9. *UnLockCurrent(queue)* – разблокирование текущего элемента очереди.

10. MAX(p1...pn) – выбор максимального значения из некоторого множества числового или множества переменных, то же самое и *MIN*(*p*1...*pn*), например:

(*fRes66: = min*(*fRes55;fRes66;fRes102;fRes104;fRes106;fRes108;fRes110; fRes112;fRes114;fRes58;*)).

3.2.3. Информационное обеспечение

Система управления БД реализуется средствами MS SQL Server. Основными таблицами БД являются: ресурсы; оборудование; персонал; преобразователи; сообщения; заявки; цели; страты; агенты; предметные области; модели; справочник типов ресурсов; связи элементов; диаграммы; входы преобразователя; выходы преобразователя; средства преобразователя.

3.2.4. Программное обеспечение

Разработанный пакет прикладных программ состоит из следующих основных модулей:

- модуль конструктора фрейм-систем;
- построение ситуационной модели МППР;
- проведение ситуационных экспериментов;
- формирование отчетов по структуре модели;
- формирование отчетов по результатам экспериментов.

3.2.5. Алгоритмическое обеспечение

Ниже перечислены основные алгоритмы, реализованные в системе BPsim2: имитатор (машина вывода); работа интеллектуального агента; диагностирование ситуации и поиск решения; декомпозиция процесса преобразования / цели; визуальное отображение связей между элементами модели; сохранение модели в базе данных; загрузка модели из БД; копирование модели; отображение динамических параметров модели в витрине данных; проведение функционально-стоимостного анализа

статической модели; анимация состояния процесса верхнего уровня на основе состояний процессов нижнего уровня.

Наиболее сложными являются алгоритмы работы ИА и имитатора. За основу алгоритма имитатора взят алгоритм машины вывода, описанный в разделе 2.5.2. На рис. 3.1 приводится блок-схема работы имитатора.

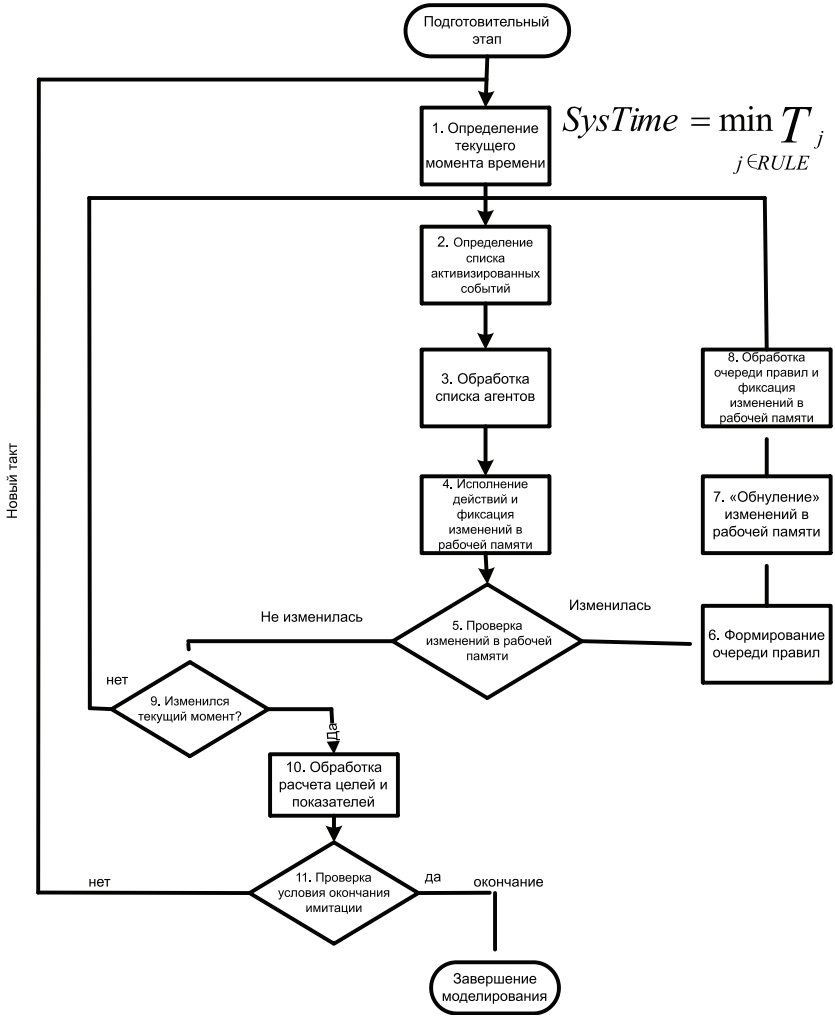


Рис. 3.1. Блок-схема работы имитатора

На 4-м и 8-м шагах работы имитатора происходит формирование файла, содержащего перечень правил и выполняемых действий над объектами рабочей памяти, а также состояния характеристик объектов. Сформированный файл в дальнейшем может быть экспортирован в MS Excel. На 10-м шаге работы имитатора выполняется расчет целей, производных и консолидированных показателей.

3.2.6. Требования к аппаратному и программному обеспечению

Требования к аппаратному обеспечению, минимальная конфигурация:

– процессор Pentium 200 (рекомендуемый Pentium IV 3200);
– оперативная память не менее 512 Мб,
– монитор – 17″ (рекомендуемый 21″),
– свободное дисковое пространство – не менее 50 Мб.

Требования к программному обеспечению:

– операционная система Windows 9x и выше, Windows XP и выше,
– система управления базами данных MS SQL Server 2000,
– средство анализа данных MS Excel; MS Project.

3.3. Технология работы с системой

3.3.1. Основные этапы работы с СДМС BPsim2

Последовательность работы представлена на рис. 3.2.

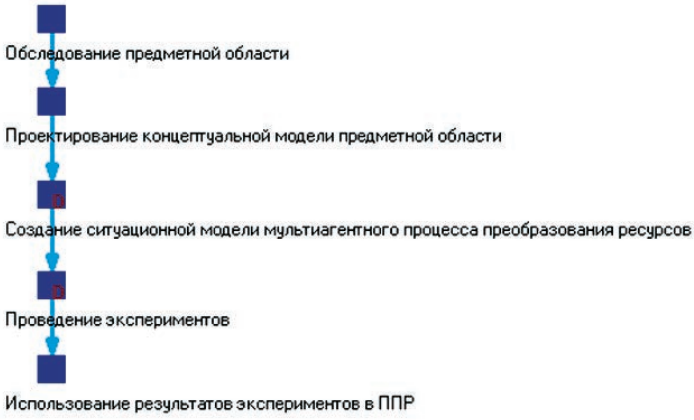


Рис. 3.2. Работа с СДМС BPsim2

Процесс состоит из следующих этапов:

1. Проектирование концептуальной модели предметной области.
2. Создание ситуационной модели мультиагентного процесса преобразования ресурсов.
3. Проведение экспериментов с ситуационной моделью мультиагентного процесса преобразования ресурсов.

3.3.2. Реализация механизма логического вывода на основе языка Transact-SQL

Как отмечено в работе [6], «чтобы системы с базами знаний, основанные на фреймовых системах, извне выглядели интеллектуальными, при их проектировании необходимо предусматривать применение в составе системы присоединенных процедур. Иначе их трудно будет отличить от обычных программ обработки данных». В данном разделе решается задача обоснования реализации механизма логического вывода на основе языка Transact-SQL.

Рассмотрим решение примера «классической» задачи на основе подхода, изложенного в разделе 2.4.2. Пример описан в [119] и демонстрирует следующие аспекты фреймов: значения по умолчанию, демоны, множественное наследование.

В данном примере рассматривается предметная область оценки недвижимости – необходимо провести оценку примерной стоимости на рынке земельных участков, полная информация о которых отсутствует. Большинство участков имеет, как правило, форму выпуклых прямоугольников, поэтому можно оценить стоимость участков, предполагая, что те, о которых идет речь, также имеют подобную форму, если только нет конкретной информации об обратном. Предположим, что цепочка «КВАДРАТ → ПРЯМОУГОЛЬНИК → ПАРАЛЛЕЛОГРАММ → ТРАПЕЦИЯ → ЧЕТЫРЕХУГОЛЬНИК → МНОГОУГОЛЬНИК» представляет знания о плоских геометрических фигурах, которые можно использовать для логических рассуждений о форме участков. Каждый узел в этой цепи имеет связанную с ним структуру записей (фрейм), формат которой приведен ниже [119].

NAME (ИМЯ):
Number of sides (Количество сторон):
Length of sides (Длины сторон):
Size of Angles (Углы):
Area (Площадь):
Price (Цена):

Практически все слоты фрейма *Многоугольник* придется оставить незаполненными, поскольку ничего нельзя сказать о сторонах и углах типичного многоугольника. Однако для слота *Количество сторон* в качестве значения по умолчанию можно установить 4, поскольку подавляющее большинство земельных участков имеет форму четырехугольника. Таким образом, все земельные участки, информация о форме контура которых отсутствует, будут полагаться четырехугольными. Слот *Площадь* также нельзя заполнить, но известно, как вычислить площадь многоугольника, располагая другой информацией о нем. Любой *n*-сторонний многоугольник можно разбить на $n - 2$ треугольника, вычислить их площади и затем просуммировать результаты. Программу, реализующую эту процедуру, можно подключить к слоту *Площадь*. Процедуры, подключенные к структуре данных и запускаемые на выполнение при появлении запроса или обновлении информации в структуре, называют *демонами*. Те демоны, которые по запросу вычисляют некоторые значения, называются *демонами по требованию* (IF-NEEDED) [119].

Полезно также иметь демон, который при заполнении слота Площадь сразу вычислял бы цену участка. Эта процедура относится к другому типу демонов – *демонам добавления* (IF-ADDED) – и подключается также к слоту *Площадь*. Теперь при обновлении или установке значения слота *Площадь* автоматически будет вычислена цена участка, а результат будет помещен в слот *Цена*.

Перейдем к следующему уровню в иерархии фреймов. Для фрейма *Четырехугольник* совершенно очевидно нужно установить значение 4 в слот *Количество сторон*. Это значение будет наследоваться фреймами на каждом из последующих уровней иерархии. Вычислять площадь и цену всех фигур, представленных фреймами последующих уровней, можно тем же способом, что и для многоугольника. Поэтому описанные выше демоны также могут быть унаследованы всеми последующими фреймами. Но для четырехугольника можно примерно оценить площадь, даже не располагая информацией о значениях внутренних углов контура, а зная только длины сторон. Вполне приемлемые результаты можно получить с помощью следующего эвристического способа: среднюю длину стороны для одной пары противолежащих сторон умножить на среднюю длину стороны для другой пары. Этот метод даст существенную ошибку только для четырехугольников, не являющихся выпуклыми, а такое встречается очень редко [119].

Эта эвристика может быть реализована в виде демона по требованию, подсоединенного к слоту *Площадь* фрейма *Четырехугольник*. Такой демон должен выполнять следующее:

– если имеется информация о величинах углов четырехугольника и длинах сторон, то необходимо вызывать демон фрейма *Многоугольник* и выполнять точное вычисление площади;

– если имеется только информация о длинах сторон четырехугольника, то следует выполнять вычисление по приближенному эвристическому методу;

– если отсутствует любая информация о параметрах четырехугольника, никаких вычислений выполнять не нужно.

Фреймы, представляющие все последующие разновидности четырехугольников, наследуют значение из слота *Количество сторон* фрейма *Четырехугольник*. Но в каждом из этих фреймов можно реализовать свою процедуру вычисления площади, лучше учитывающую особенности именно данного вида фигур. Например, площадь трапеции можно вычислить как произведение высоты на среднюю длину оснований, а фреймы прямоугольника и квадрата могут унаследовать эту процедуру у параллелограмма, площадь которого равна произведению основания на высоту [119].

Ниже приводятся листинги скриптов создания хранимых процедур, вычисляющих площади параллелограмма, прямоугольника и треугольника.

```
/*-------------------------------------------------------------------------*/
/*-------------------------------------------------------------------------*/
/*
Параллелограмм – четырехугольник, у которого стороны попарно параллельны.
-------------------------------------
--вычислить площадь параллелограмма
--@a-длина одной стороны,
--@b – длина следующей стороны,
--@aa – угол между сторонами @a и @b в градусах,

--@S-результат (площадь)
*/
CREATE PROCEDURE sp_Parallelogram (@a DECIMAL(6,2), @b
DECIMAL(6,2), @aa FLOAT, @S DECIMAL(6,2) OUT)
AS
SELECT @S = @a*@b*SIN(@aa*PI()/180)
```

```
GO
/*-------------------------------------------------------------------------*/
/*-------------------------------------------------------------------------*/



/*
Прямоугольник – четырехугольник, у которого все углы прямые.
------------------
вычислить площадь прямоугольника,
        где@a – длина одной стороны,
        @b – длина следующей стороны,
        @s – результат, площадь.
*/
CREATE PROCEDURE sp_Restangle (@a DECIMAL(6,2), @b FLOAT, @S
DECIMAL(6,2) OUT)
AS
SELECT @S = @a*@b
GO
/*-------------------------------------------------------------------------*/
/*-------------------------------------------------------------------------*/
/*
Треугольник – многоугольник с тремя сторонами.
--
площадь треугольника
        @a, @b – 2 смежные стороны
        @ab – угол между ними, в градусах

        @s-результат
*/
CREATE PROCEDURE sp_Treangle (@a DECIMAL (6,2), @b DECIMAL (6,2),
@ab DECIMAL(6,2), @s DECIMAL (6, 2) OUT)
AS
SELECT @S = 0.5*@a*@b *sin (@ab * pi()/180)
GO

/*-------------------------------------------------------------------------*/
/*-------------------------------------------------------------------------*/
```

Ниже приводится хранимая процедура вычисления стоимости площади многоугольника:

```
/*--------------------------------------------------------------------------*/
/*--------------------------------------------------------------------------*/

CREATE PROCEDURE sp_Area (
        @TypeOfFigure VARCHAR(50), --типфигуры
@NUMBEROFSIDES SMALLINT = 4, -- КОЛИЧЕСТВО СТОРОН, ПО
    УМОЛЧАНИЮ ЧЕТЫРЕХУГОЛЬНИК
        @LengthOfSides VARCHAR(255) = '', --длины сторон, разделенные
спец. символом '!' в м2, указываются в последовательности друг за другом
                                --н-р '6.2!7.8!8.1!9.3!'
        @SizeOfAngless VARCHAR(255) = '', --углы в градусах, начиная с угла
между первой и второй стороной указанной в параметре @LengthOfSides
                                --н-р, '30!40!50!'
        @StrOfTreangle VARCHAR(255) = '', --при расчете площади много-
угольника – метод разбиения на треугольники, необходимо задать
                                --стороны и угол между ними, в формате
'сторона1:сторона2:угол между ними:!'сторона1:сторона2:угол между ними:!
        @PriceOfM MONEY = 0,--стоимость метра квадратного
        @Price MONEY = 0 OUT) --стоимость участка
AS
DECLARE @tblOfSides TABLE (ID INT, VAL FLOAT)
DECLARE @tblOfAngless TABLE (ID INT, VAL FLOAT)
DECLARE @tblofTreangle TABLE (ID INT, VAL VARCHAR(100), state
TINYINT)
DECLARE @Area FLOAT, @a FLOAT, @b FLOAT, @S DECIMAL (6,2), @aa
FLOAT
DECLARE @cnt INT, @id INT
SELECT @Area = 0, @Price = 0, @S = 0
SELECT @cnt = 0

--процедура формирования из строки – табличного представления

INSERT INTO @tblOfSides ([id], val)
        SELECT* FROM dbo.FromStrToTable(@LengthOfSides, '!')

INSERT INTO @tblOfAngless ([id], val)
```

```
        SELECT* FROM dbo.FromStrToTable(@SizeOfAngless, '!')

IF (@NumberOfSides > 4) BEGIN
        INSERT INTO @tblOfTreangle (id, val)
        SELECT * FROM dbo.FromStrToTable (@StrOfTreangle, '!')
        UPDATE @tblOfTreangle SETState = 0

        SELECT @cnt = COUNT(*) FROM @tblOfTreangle WHEREState = 0
        WHILE @cnt > 0 BEGIN
                DELETE FROM @tblOfSides
                SELECT TOP 1 @id = id, @LengthOfSides = Val FROM @
tblOfTreangle WHEREState = 0
                 INSERT INTO @tblOfSides
                        SELECT* FROM dbo.FromStrToTable(@LengthOfSides, ':')
                SELECT @a = (SELECT Val FROM @TblOfSides WHERE ID = 1)
                SELECT @b = (SELECT Val FROM @TblOfSides WHERE ID = 2)
                SELECT @aa = (SELECT Val FROM @TblOfSides WHERE ID = 3)
                EXEC sp_Treangle @a, @b, @aa, @s OUT

                SELECT @cnt = COUNT(*) FROM @tblOfTreangle WHEREState = 0
                SELECT @Area = @Area + @s
                UPDATE @tblOfTreangle SETState = 1 WHERE id = @id
        END
END
IF (@NumberOfSides = 4) BEGIN
        --если тип фигуры не определен, и углы неизвестны, но известны дли-
ны -- > то общая формула для четырехугольника:
        --средняя длина стороны для одной пары противолежащих сторон ум-
ножить на среднюю длину стороны для другой пары
IF (UPPER(@TypeOfFigure) = 'NONAME') AND ((SELECT COUNT(*) FROM
@TblOfSides) = 4) AND (SELECT COUNT(*) FROM @TblOfAngless) = 0
BEGIN
                SELECT @Area = (SELECT AVG(VAL) FROM @TblOfSides WHERE
ID IN (1, 3))* (SELECT AVG(VAL) FROM @TblOfSides WHERE ID IN (2, 4))
        END
        --квадрат
        IF (UPPER(@TypeOfFigure) = 'квадрат') AND ((SELECT COUNT(*)
FROM @TblOfSides) > = 1) BEGIN
                SELECT @a = (SELECT Val FROM @TblOfSides WHERE ID = 1)
```

```
            EXEC sp_Restangle @a, @a, @Area OUT
        END
        --прямоугольник
        IF (UPPER(@TypeOfFigure) = 'прямоугольник') AND ((SELECT
COUNT(*) FROM @TblOfSides) > = 1) BEGIN
            SELECT @a = (SELECT Val FROM @TblOfSides WHERE ID = 1)
            SELECT @b = (SELECT Val FROM @TblOfSides WHERE ID = 2)
            EXEC sp_Restangle @a, @b, @Area OUT
        END
        --трапеция
        IF (UPPER(@TypeOfFigure) = 'Параллелограмм') AND ((SELECT
COUNT(*) FROM @TblOfSides) > = 2) AND ((SELECT COUNT(*) FROM @
TblOfSides) > = 1) BEGIN
            SELECT @a = (SELECT Val FROM @TblOfSides WHERE ID = 1)
            SELECT @b = (SELECT Val FROM @TblOfSides WHERE ID = 2)
            SELECT @aa = (SELECT Val FROM @TblOfAngless WHERE ID = 1)
            EXEC sp_Parallelogram @a, @b, @aa, @Area OUT

END
        --трапеция
        IF (UPPER(@TypeOfFigure) = 'ромб') AND ((SELECT COUNT(*)
FROM @TblOfSides) > = 1) AND ((SELECT COUNT(*) FROM @
TblOfSides) > = 1) BEGIN
            SELECT @a = (SELECT Val FROM @TblOfSides WHERE ID = 1)
            SELECT @aa = (SELECT Val FROM @TblOfAngless WHERE ID = 1)
            EXEC sp_Parallelogram @a, @a, @aa, @Area OUT
        END
END
IF @Area > 0 BEGIN
        SELECT @Price = @Area * @PriceOfM
END
SELECT Price = @Price
GO

/*---------------------------------------------------------------*/
/*---------------------------------------------------------------*/
```

На рис. 3.3–3.5 приводятся примеры вычисления стоимости участков различной формы в QueryAnalyzer (утилита MSSQLServer):

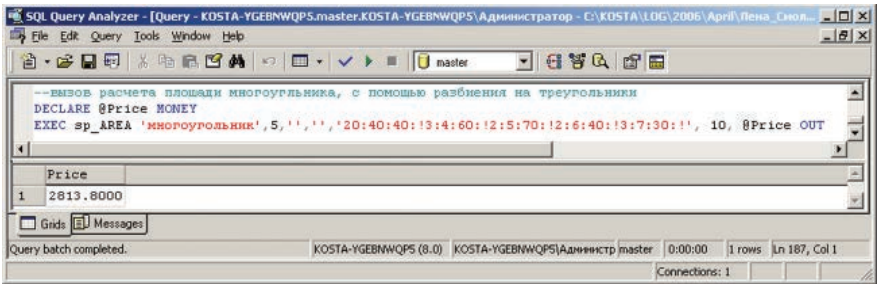


Рис. 3.3. Вызов расчета стоимости площади многоугольника, с помощью разбиения на треугольники

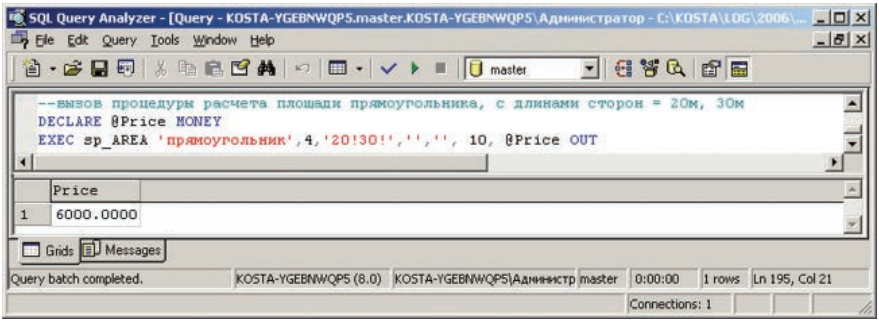


Рис. 3.4. Вызов расчета стоимости площади прямоугольника

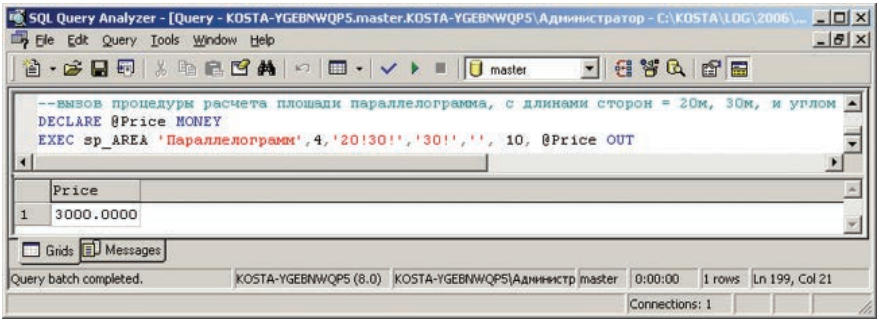


Рис. 3.5. Вызов расчета стоимости площади параллелограмма

В целом можно выделить следующие варианты применения языка Transact-SQL для реализации функции логического вывода в мультиагентной системе преобразования ресурсов:

– механизм вывода (алгоритм) ИА полностью или частично оформлен в виде хранимой процедуры и каждый раз вызывается планировщиком;

– части правил ИА содержат или запросы к БЗ на языке Transact-SQL, или ссылки на хранимые процедуры, которые реализуют функции поиска и/или расчетов (вычислений);

– решение задач поиска и вычислений на фрейм-системе (так как модели реальных МАС преобразования ресурсов имеют большую размерность).

Выводы по разделу 3.3.2:

1. Разработан подход к созданию фрейм-системы на основе реляционной БД.

2. Достоинством предложенного решения является то, что проектирование модели предметной области в виде фреймовой системы, построение концептуальной модели предметной области, ввод знаний и данных, механизм логического вывода, и запросы к базе знаний реализуются на языке Transact-SQL, т. е. не потребовалась разработка языка вывода на фреймовой модели. Данный фактор снижает требования к навыкам системных программистов, аналитиков и инженеров по знаниям, поддерживающих работоспособность данной системы, а также автоматизирует их работу. Использование промышленной СУБД MS SQL Server для хранения данных базы знаний позволяет интегрировать данную систему с корпоративной системой предприятия.

3.3.3. Описание фреймовой оболочки экспертных систем «Конструктор фрейм-систем»

Разработанная оболочка ЭС «Конструктор фрейм-систем» [142] поддерживает следующие функции:

1) конструирует фреймовые модели, позволяющие хранить информацию о предметной области;

2) проектирует КМПО в виде фрейм-концептов и концептуальных графов (см. раздел 2.4.2);

3) модифицирует данные о предметной области;

4) изменяет структуру данных о предметной области;

5) поддерживает операции обработки данных и логического вывода при любой допустимой структуре фреймовой модели, с использованием языка Transact-SQL;

6) изменяет структуру данных под воздействием своих данных.

Главная форма оболочки и форма построения SQL-запросов представлена на рис. 3.6 и 3.7.

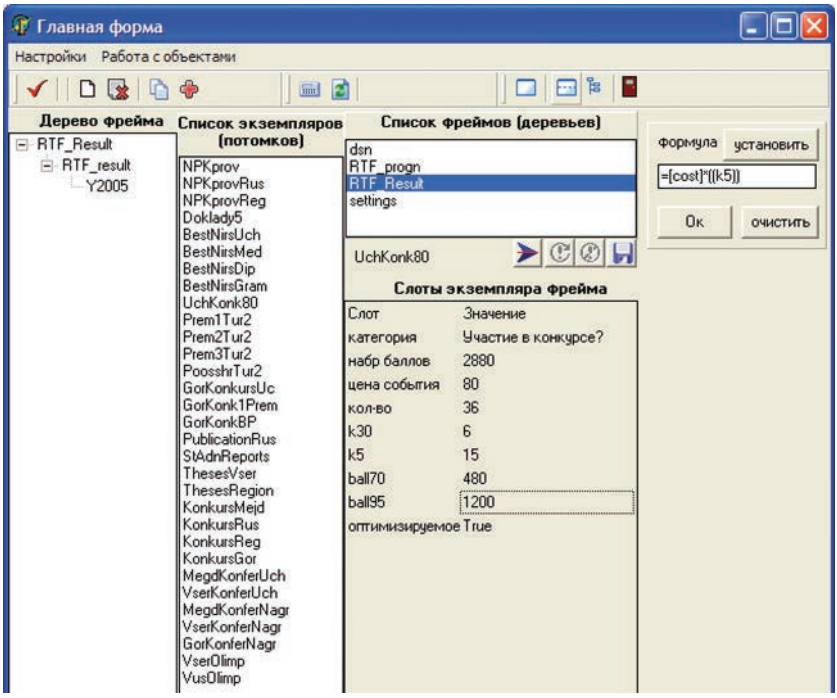


Рис. 3.6. Главная форма оболочки ЭС

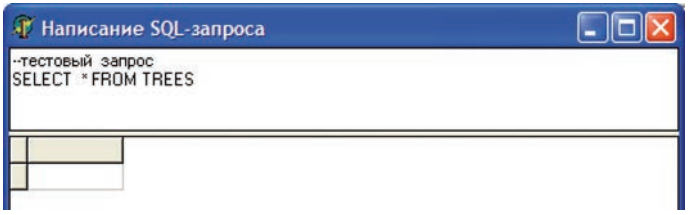


Рис. 3.7. Форма построения SQL-запроса к фреймовой системе

3.3.4. Создание ситуационной моделипроцессов преобразования ресурсов

За основу построения СДМС была взята проблемно-ориентированная система имитационного моделирования (СИМ) BPsim [136]. Подобный подход к созданию СДМС был применен А.Ю. Филипповичем [18], когда СДМС была построена (расширена) на базе проблемно-ориентированной СИМ допечатных процессов [19]. Последовательность построения мультиагентной ситуационной модели процесса преобразования ресурсов включает следующие этапы (рис. 3.8).

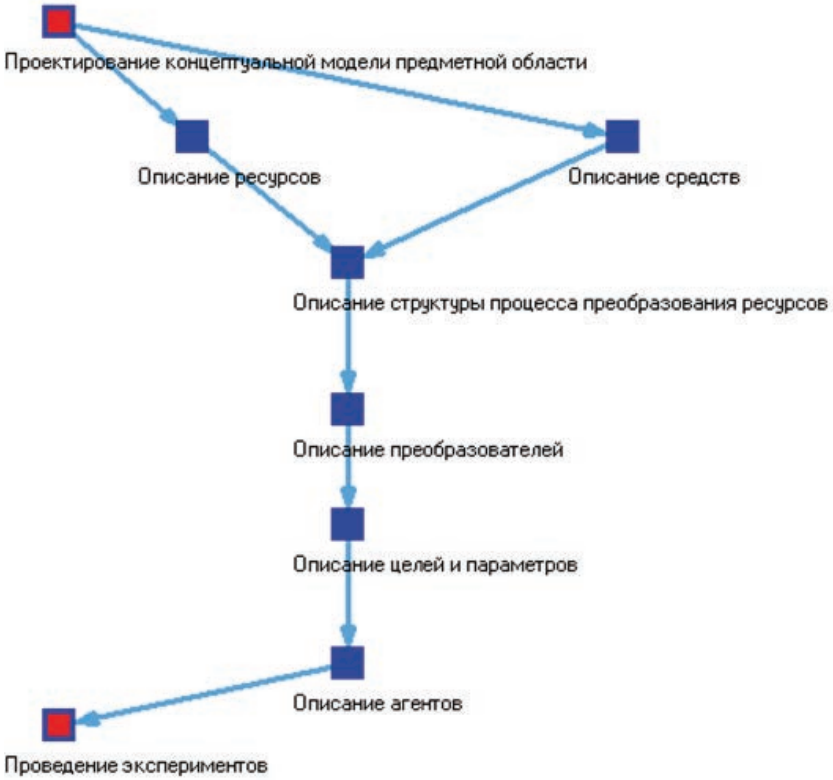


Рис. 3.8. Этапы построения ситуационной модели мультиагентных процессов преобразования ресурсов

Добавление и выбор предметной области показано на рис. 3.9–3.10.

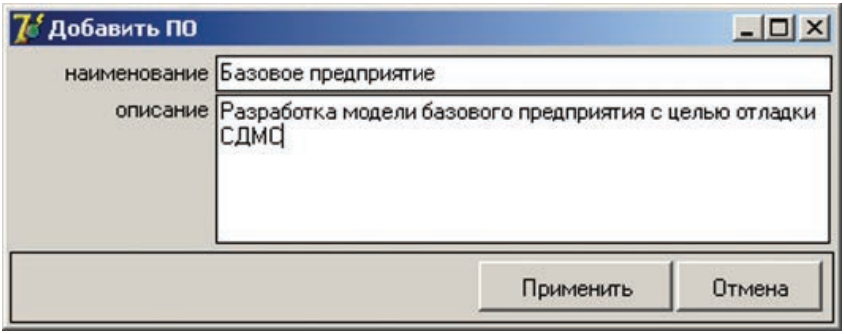


Рис. 3.9. Добавление новой предметной области

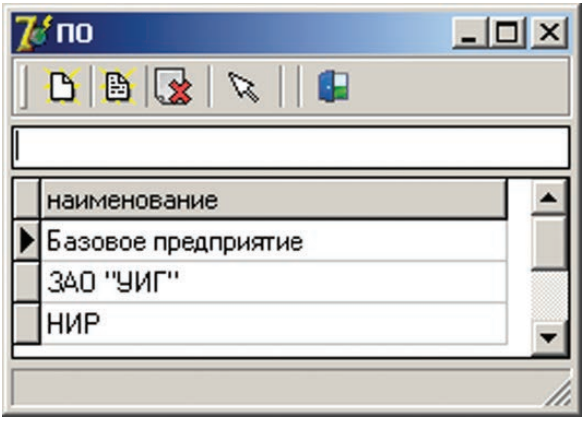


Рис. 3.10. Форма выбора предметной области

Рассмотрим этапы построения ситуационной модели мультиагентных процессов преобразования ресурсов в СДМС BPsim2:

1) обследование ресурсов и средств процесса преобразования (персонал, оборудование, сырье, материалы). Описание ресурсов процесса преобразования осуществляется через форму справочника ресурсов, а описание конкретного ресурса – с помощью формы «Описание ресурса» (рис. 3.11). Описание персонала и оборудования осуществляется через формы «Карточка работника» (рис. 3.12) и «Описание оборудования» (рис. 3.13) соответственно.

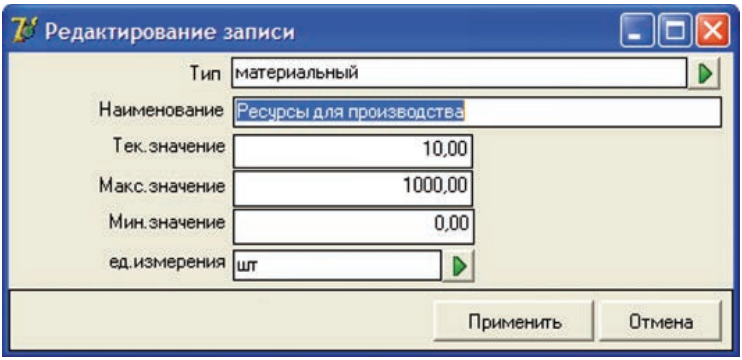


Рис. 3.11. Форма описания ресурса

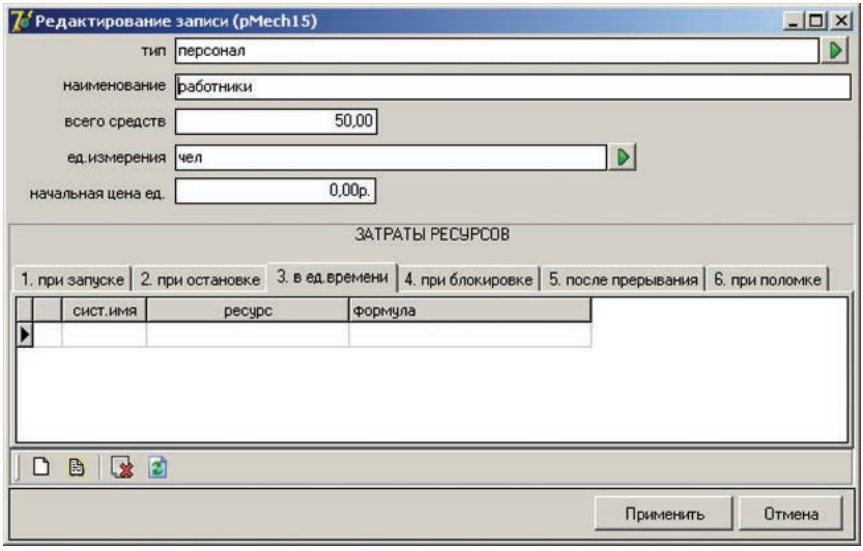


Рис. 3.12. Карточка работника

2) моделирование процессов – проектирование внутренней структуры процессов преобразования ресурсов. На рис. 3.14 приведен пример описания процесса в BPsim2.

Через интерфейс описания операции (рис. 3.15 и 3.16) описывается захват и освобождение ресурсов, механизмов, формирование продуктов, длительность преобразования, приоритет.

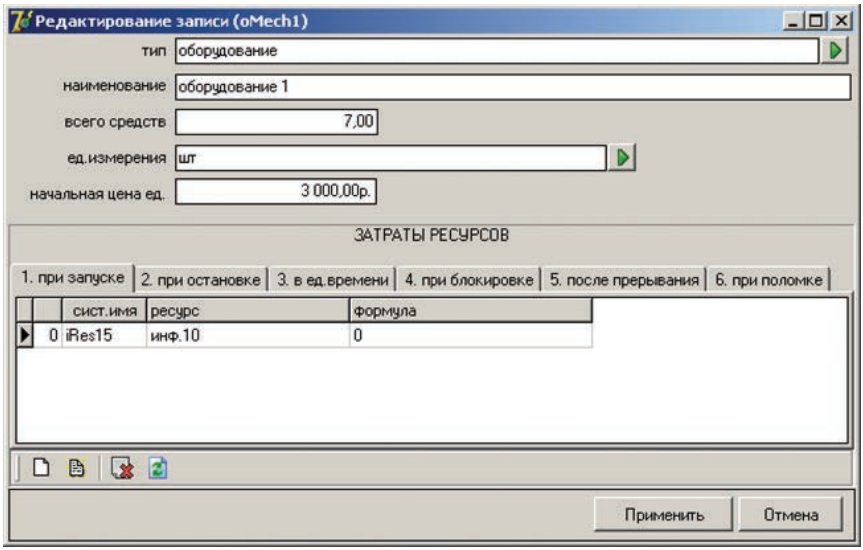


Рис. 3.13. Описание средства

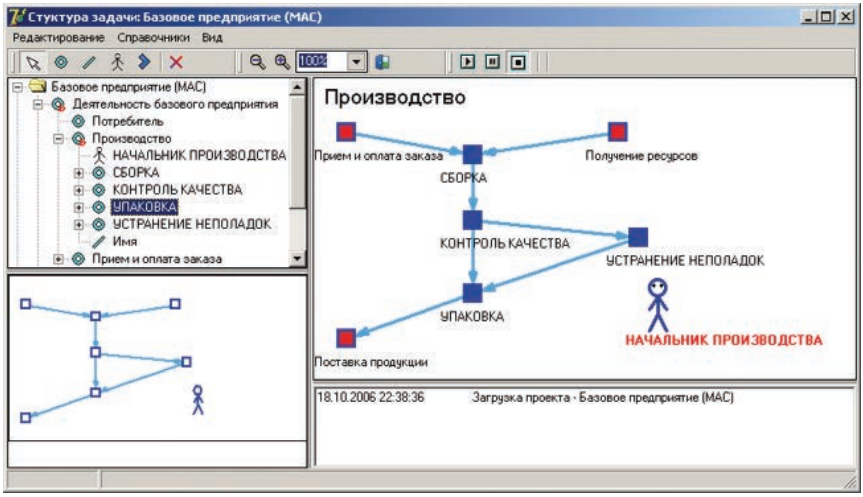


Рис. 3.14. Визуальное представление процесса в СДМС

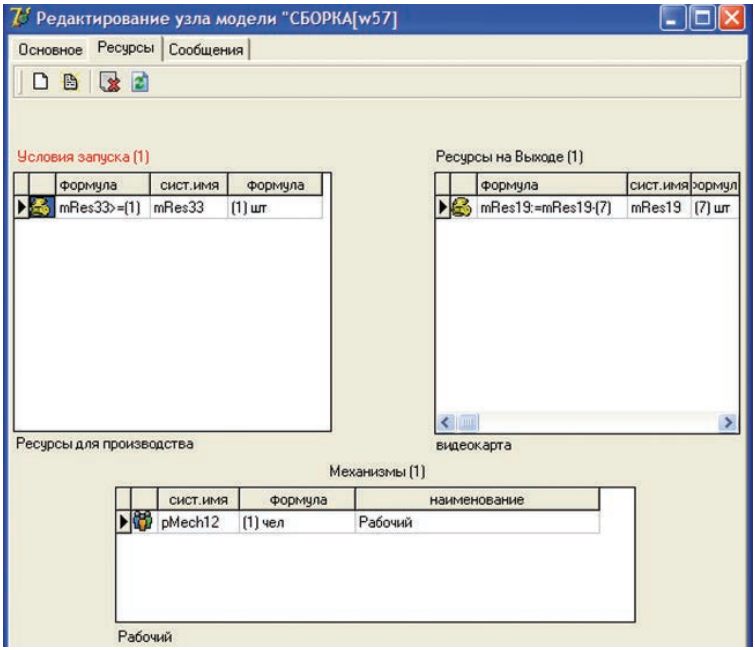


Рис. 3.15. Форма описания входов, выходов преобразователя

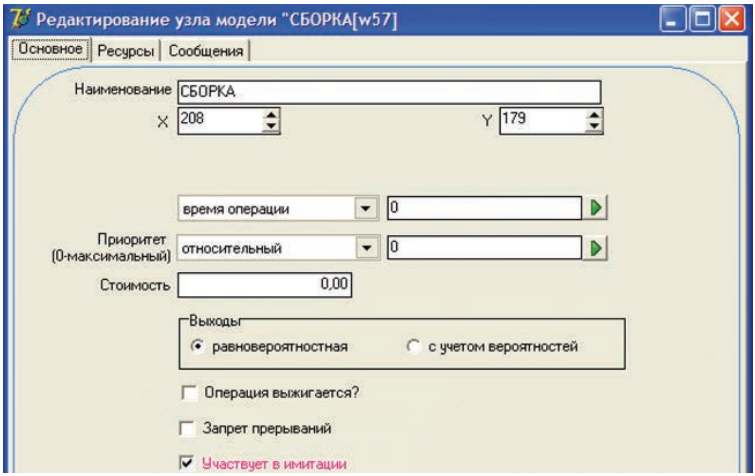


Рис. 3.16. Описание преобразователя

На рис. 3.17–3.19 представлены формы описания модели агента: целей, базы знаний и ситуаций соответственно.

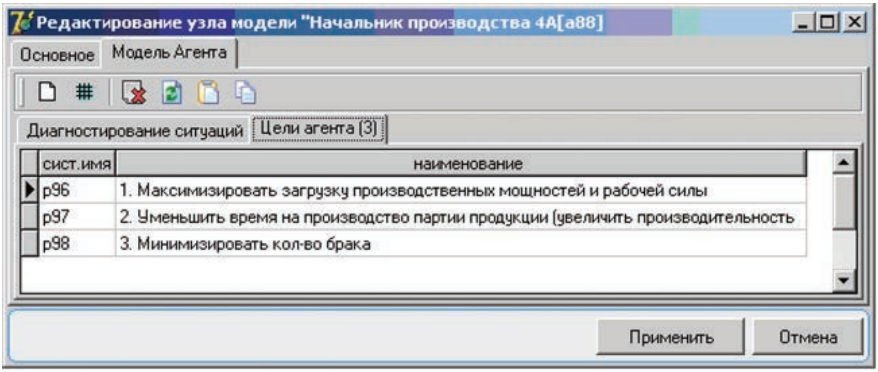


Рис. 3.17. Описание целей агента

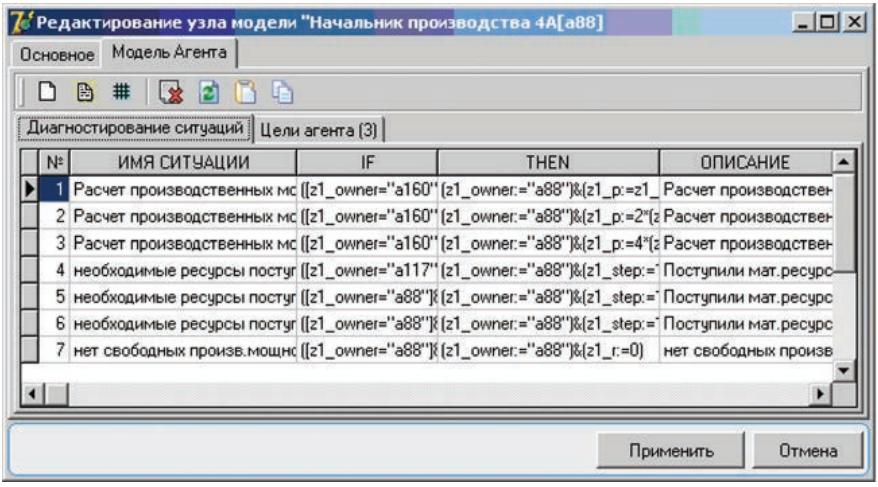


Рис. 3.18. Описание базы знаний агента

Описание части IF осуществляется посредством вызова стандартной формы «Формула» – рис. 3.20 или панели описания выражений.

Форма описания части THEN правила агента представлена на рис. 3.21.

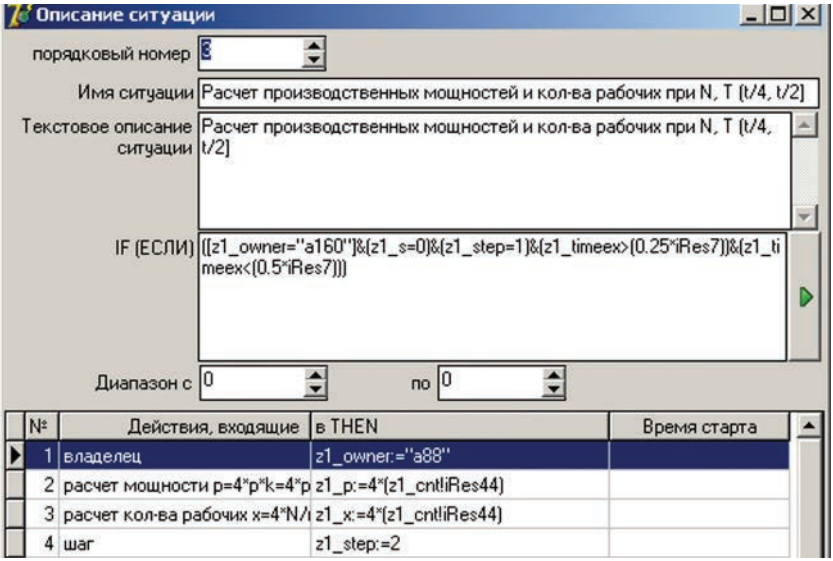


Рис. 3.19. Форма описания ситуации

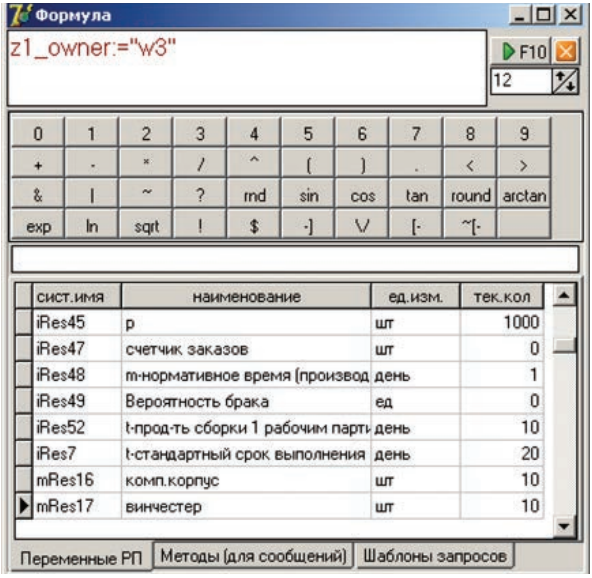


Рис. 3.20. Форма«Формула»

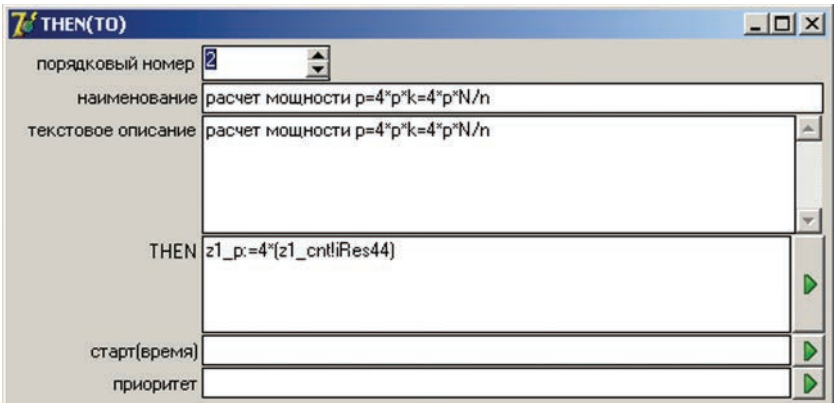


Рис. 3.21. Форма описания части THEN правила агента

На рис. 3.22–3.25 представлены формы описания модели целей.

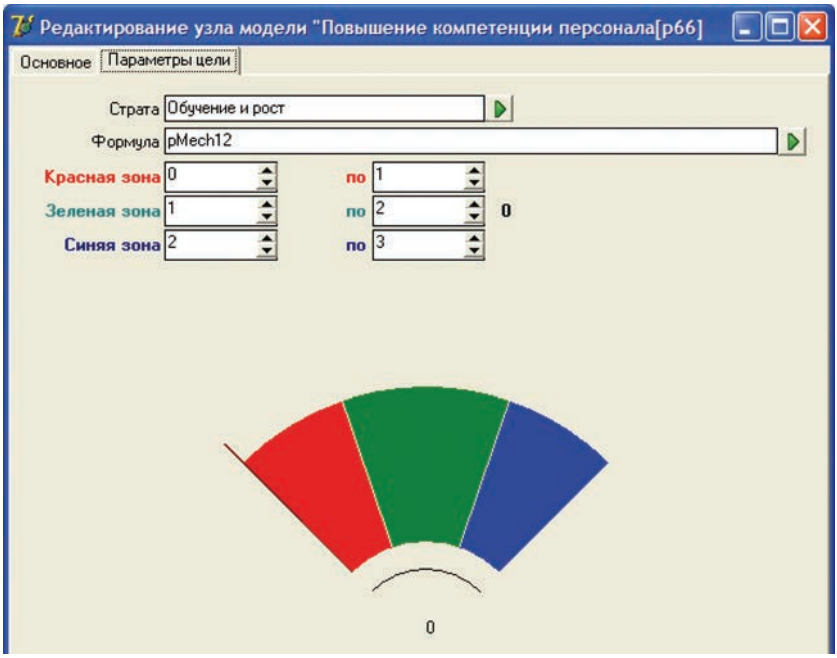


Рис. 3.22. Описание цели

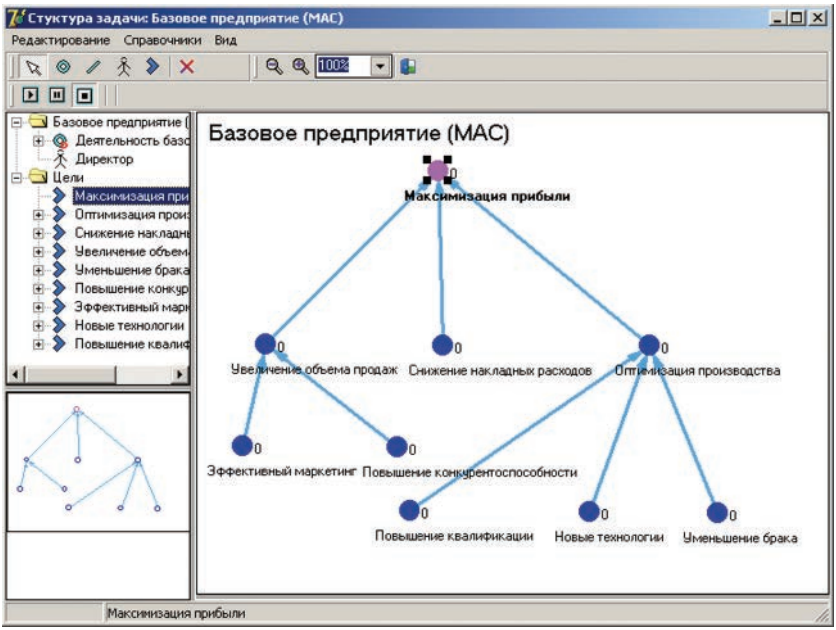


Рис. 3.23. Описание дерева целей

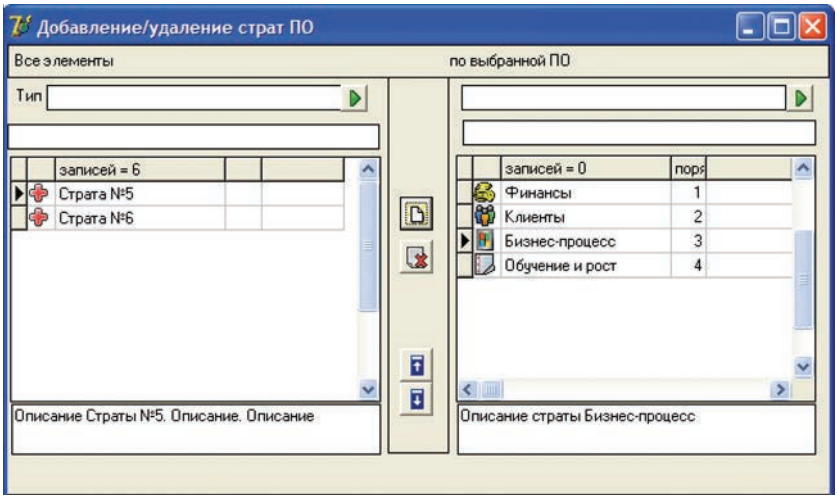


Рис. 3.24. Форма выбора страт для данной предметной области

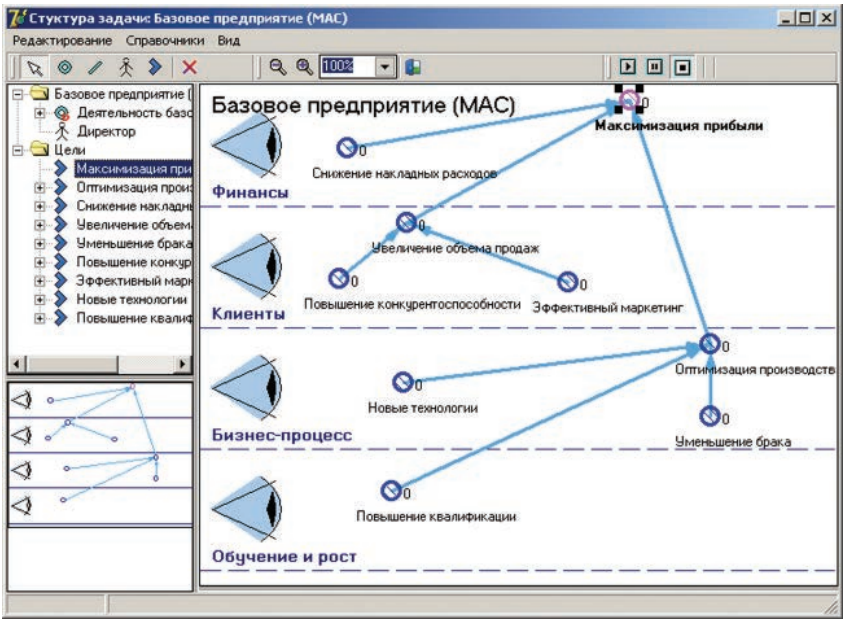


Рис. 3.25. Графическое представление целей
в системе сбалансированных показателей BSC

Интеграция СДМС и автоматизированного инструментария системы сбалансированных показателей усиливает проблемную ориентацию системы моделирования в сторону систем поддержки принятия решений на стратегическом уровне, уменьшает разрыв между системным аналитиком и ЛПР и в целом минимизирует время, необходимое для принятия решения.

Внешний вид СДМС приведен на рис. 3.26.

3.3.5. Проведение экспериментов с моделью

Технологический процесс проведения экспериментов состоит из следующих этапов (рис. 3.27):

1) ввод начальных условий – установка начального состояния модели, настройка режимов ИМ;

2) проведение эксперимента в соответствии с установками, сделанными пользователем на предыдущем шаге, – наблюдение эксперимента и ввод/вывод данных.

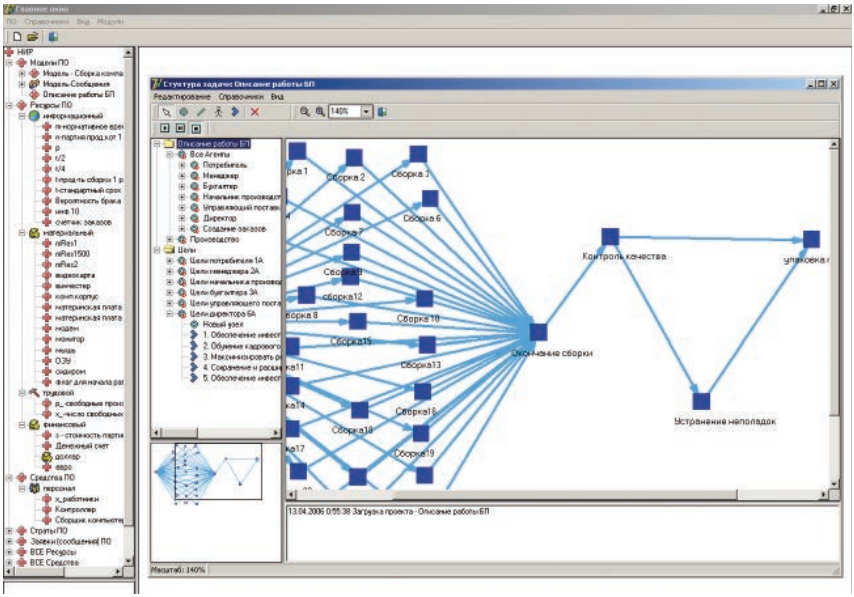


Рис. 3.26. Внешний вид СДМС

Рис. 3.27. Технологический процесс проведения экспериментов

Для описания в СДМС имитационных моделей процессов преобразования ресурсов используется механизм иерархических графов. При проведении экспериментов с ситуационной моделью мультиагентного процесса преобразования ресурсов пользователь получает доступ к просмотру Календаря событий (рис. 3.28) и Рабочей памяти (рис. 3.29).

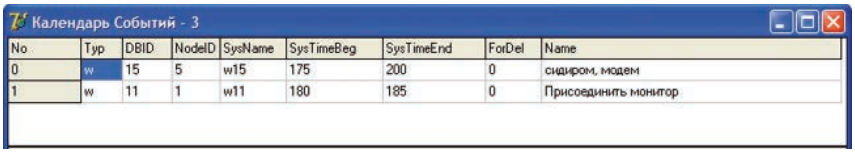


Рис. 3.28. Форма доступа к Календарю событий

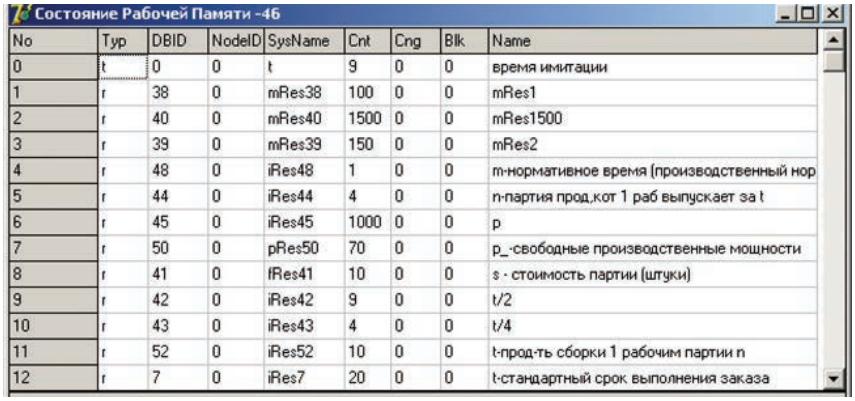


Рис. 3.29. Форма доступа к Рабочей памяти

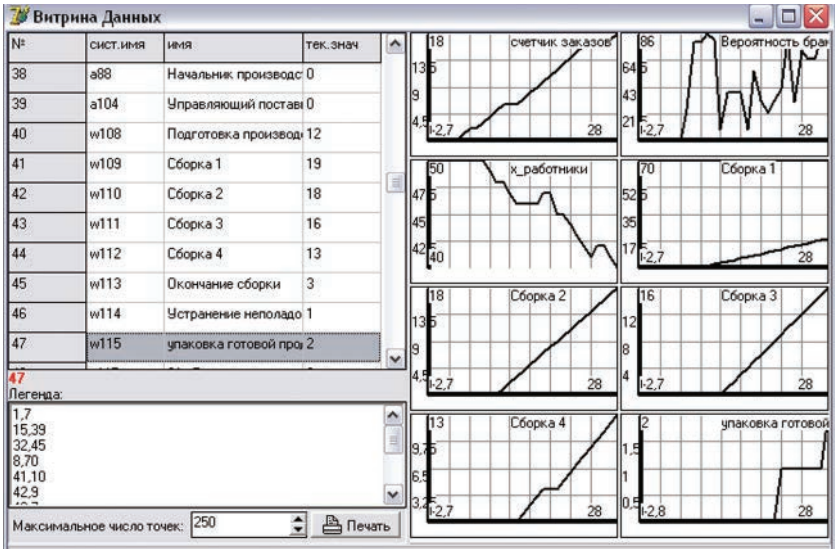


Рис. 3.30. Витрина данных

3) анализ результатов – получение отчета с результатами эксперимента и их анализ. Анализ средствами BPsim2 (рис. 3.30). В случае необходимости возможен переход на следующий шаг;

4) корректировка модели – модель корректируется и весь цикл повторяется заново до тех пор, пока не будет построена адекватная модель предметной области или не найдено решение.

Выводы

1. На основе ситуационной математической модели мультиагентного процесса преобразования ресурсов, предложенной во второй главе, были разработаны:

- подход к созданию фрейм-системы на основе реляционной БД. Достоинством предложенного решения является то, что проектирование модели предметной области в виде фреймовой системы, построение концептуальной модели предметной области, ввод знаний и данных, механизм логического вывода, и запросы к базе знаний реализуются на языке Transact-SQL;
- интерфейсы СДМС, ориентированные на конечного пользователя;
- программное, информационное, алгоритмическое, методическое обеспечение СДМС «BPsim2»;
- технология работы с системой «BPsim2».

2. Разработанная система BPsim2 обладает полным перечнем функциональных возможностей, предъявляемых к проблемно-ориентированной СДМС, и отличается:

- полным набором функциональных возможностей мультиагентной СДМС;
- наличием средств проектирования концептуальной модели предметной области;
- эффективными средствами моделирования конфликтов на общих ресурсах и средствах;
- возможностью построения мультиагентных моделей, содержащих интеллектуальные агенты;
- понятийным аппаратом, ориентированным на проблемную область мультиагентных процессов преобразования ресурсов;
- интеграцией с методикой BSC;
- **п**оддержкой русского языка.

4. ПРИМЕНЕНИЕ СИСТЕМЫ

4.1. Базовое предприятие

Модель описывает деятельность абстрактного предприятия, которое занимается производством некоторого продукта. С помощью данной модели проводится сравнительный анализ инструментальных средств имитационного моделирования мультиагентных систем с разработанным пакетом BPsim2.

Предприятие работает по схеме «производство на заказ». Рассматривается производство одного вида продукции; для этого производства требуется один вид ресурса. Запасы необходимых ресурсов для производства и запасы готовой продукции на складе отсутствуют.

На основе заказа от потребителя вычисляется количество ресурсов, необходимых для производства, формируется заказ поставщику. После получения требуемого количества ресурсов начинается производство партии продукции. При описании деятельности предприятия используется понятие бизнес-процесс, который можно представить в виде процесса преобразования ресурсов.

Процессы ресурсного взаимодействия предприятия с внешней средой определяют состав внутренних процессов предприятия. В этой среде выделяем такие объекты, как конкуренты и потенциальные потребители товаров и услуг, производимых базовым предприятием. Для формализации во внешней среде выделим таких агентов, как «Потребитель» и «Потенциальный потребитель».

Во внутренней среде предприятия выделяем следующих агентов: менеджер, начальник производства, бухгалтер, снабженец, директор. У каждого агента свои цели и свой сценарий поведения. Одни цели являются общими для ряда агентов, другие вынуждают их конфликтовать между собой.

Схема взаимодействия агентов представлена на рис. 4.1.

Обозначения в модели следующие:

n – партия продукции, которую 1 рабочий выпускает за время t (это наименьшая единица измерения продукции);

t – время выпуска одной партии 1-м рабочим;

s – стоимость одной партии;

p – производственные мощности для выпуска 1 партии 1 рабочим за время t;

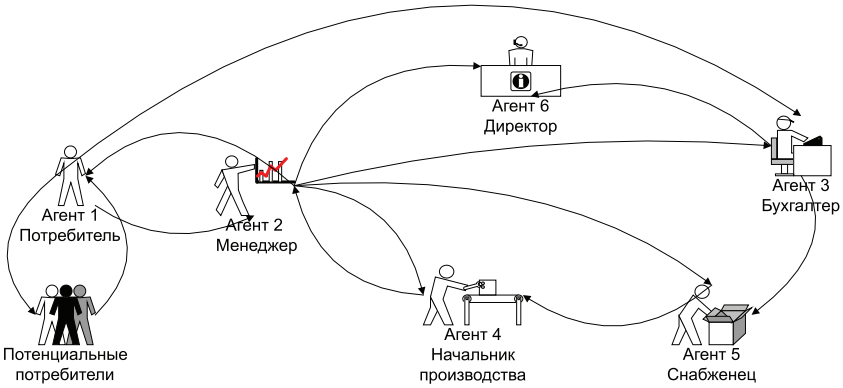


Рис. 4.1. Схема взаимодействия агентов в модели

S_i – стоимость i-го заказа;

N_i – кол-во продукции i-го заказа, $N_i = k_i n$;

k_i – кол-во заказанных партий i-го заказа, целое число;

T_i – время выполнения i-го заказа;

H_i – сумма штрафных санкций i-го заказа;

z_i – i-й заказ;

$S_i рес$ – стоимость ресурсов i-го заказа ($S_i рес = 0{,}1S_i$);

$S_i раб$ – стоимость рабочей силы i-го заказа ($S_i раб = 0{,}2S_i$);

$S_{i\,\text{накл}} = 0{,}5S_i$ – накладные расходы (заработная плата ИТР – инженерно-технических работников, АУП – административно- управленческого персонала, коммунальные услуги, ремонты, информационные услуги, и т. д.), налоги, оборудование, аренда здания и т. д.

$prib_i = S_i - S_{i\,\text{пр}} - S_{i\,\text{накл}} - h_i$ – прибыль i-го заказа;

x_i – кол-во основных рабочих для выполнения заказа $z_{i;}$

y_i – дополнительные рабочие, привлеченные к заказу $z_{i;}$

T_i^* – фактическое время изготовления заказа $z_{i;}$

p_i – производственные мощности для $z_{i;}$

S^* – предельная стоимость заказа, которую готов потратить потребитель 1А;

U^* – предельное время задержки выполнения заказа z_i, которое готов ждать потребитель 1А;

τ_i – кол-во дней просрочки выполнения заказа $z_{i;}$

b_i – кол-во брака при выполнении заказа z_i;

S_i пр – прямые затраты (S_i пр = S_i рес + S_i раб = $0{,}1S_i + 0{,}2S_i + \Delta^* y_i$);

i – номер заказа (на начало года $i = 0$);

r_j – признак состояния заказа z_i («0» – не рассчитанный, «4» – рассчитанный, «1» – идущий, «2» – завершенный, «3» – отказ потребителя);

μ – нормативное время (предпроизводственный норматив, 5 дней);

x^* – число свободных основных рабочих, $x^* = PR - \sum_{j=1}^{i} x_j$ (первоначально $x^* = PR$);

p^* – свободные производственные мощности, $p^* = ST - \sum_{j=1}^{i} p_j$ (первоначально $p^* = ST$), здесь j – бегущий индекс по идущим заказам;

PR – общее количество рабочих на предприятии;

ST – кол-во производственных мощностей на предприятии;

Θ – текущее время в днях (по абсолютной шкале времени от 0 до 365 дней, первоначально 0);

Θ_i – время начала выполнения заказа $z_{i;}$

$B[365]$ – массив плановой загрузки производственных мощностей, в начале года $B = 0$. Элемент $B[\Theta]$ показывает, сколько производственных мощностей занято в день Θ по плану.

Δ – единовременные затраты на поиск одного дополнительного рабочего;

U_i – время задержки выполнения заказа $z_{i,}$ время на ожидание освобождения производственных мощностей;

Discard_Time – время, через которое Потребитель, уже купивший товар, снова становится Потенциальным потребителем;

-*Ad_Effectiveness* – эффективность от рекламы;

-*Contact_Rate* – число контактов данного Потребителя с другими в день;

-*Adoption_Fraction* – сила убеждения Потребителя.

Описание моделей агентов базового предприятия приведено в таблице А.1 приложения.

Ограничения модели

1. Моделируется процесс организации базового предприятия, завоевания им рынка и создания своей клиентской базы. Поэтому вначале все потребители – потенциальные.

2. Количество привлеченных рабочих не больше количества рабочих на всем предприятии ($y \le$ PR). Предполагаем, что при нехватке рабочей силы «Начальник производства» обращается на рынок труда и привлекает недостающих рабочих. При этом предприятие несет дополнительные затраты в размере Δ на каждого рабочего.

3. Количество привлеченных к заказу станков (производственных мощностей) не больше общего количества станков за вычетом количества станков, находящихся в ремонте. Обозначим наибольшее доступное кол-во станков ST ($p \le ST$). Ограничение по производственным мощностям является самым существенным, т. к. при их нехватке задерживается выполнение заказа и предприятие платит штрафы.

4. «Управляющий поставками» не заказывает ресурсы у поставщиков, пока «Потребитель» не перечислил деньги за заказ.

5. «Начальник производства» не начинает выполнение заказа, пока не получены материалы.

6. Продукция не отгружается «Потребителю», пока не устранен брак.

Временные показатели модели

1. Считаем, что время *t* пошло, когда «Начальник производства» запустил заказ в производство. После того, как «Потребитель» перечислил деньги проходит μ дней до фактического начала производства (даже при наличии ресурсов). Время μ – предпроизводственный норматив, например 5 дней. В это время μ (дней) входит: банковские дни; время на заказ материально – технических ресурсов (МТР); время на проплату МТР; время на доставку МТР; время на входной контроль качества МТР; время на подготовку конструкторско-технологической документации.

2. Время выполнения заказа включает: время на производство продукции; время на контроль качества готовой продукции; время на исправление брака; время на предпродажную подготовку и упаковку.

Основные процессы базового предприятия представлены на рис. 4.2.

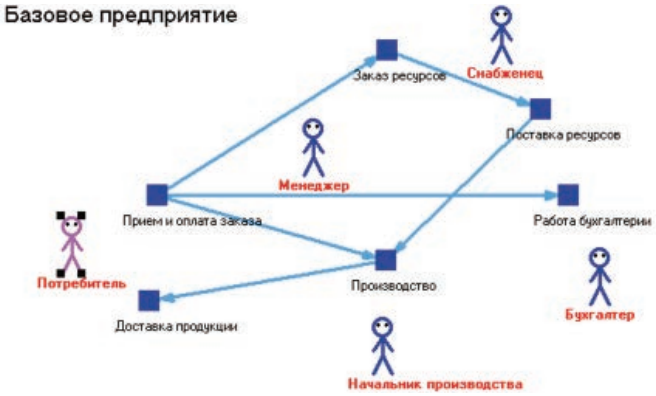


Рис. 4.2. Основные процессы базового предприятия

Исходные данные представлены в табл. 4.1.

Таблица 4.1

Исходные данные

Переменная	Наименование	Значение	Ед. изм.
t	время выпуска одной партии 1-м рабочим	4	дней
PR	общее количество основных рабочих	50	шт.
ST	количество производственных мощностей	70	шт.
μ	предпроизводственный норматив	5	дней
n	партия продукции, которую 1 рабочий выпускает за время *t*	4	шт.
p	производственные мощности для выпуска 1 партии 1 рабочим за время *t*	1	шт.
s	стоимость одной партии	30	тыс. руб
Δ	затраты на поиск одного дополнит. рабочего	0,5	тыс. руб

В данном примере решается задача обработки 20 заказов. При этом значения количества партий товара и времени выполнения были выбраны случайным образом. Основные показатели модели представлены в табл. 4.2. Временные показатели модели представлены в табл. 4.3. Диаграмма Ганта по использованию производственных мощностей представлена в табл. 4.4. Диаграмма Ганта по задействованному числу основных рабочих представлена в табл. 4.5.

Разработка модели базового предприятия и проведенные эксперименты в системе BPsim2 позволили провести отладку СДМС и дали основание перейти к решению практических задач.

Таблица 4.2

Основные показатели модели

Номер заказа	Кол-во заказанных партий	Кол-во продукции	Время выполнения заказа Z_i	Производственные мощности заказа Z_i	Стоимость заказа Z_i	Кол-во рабочих для выполнения заказа	Дополнительные рабочие, привлеченные к заказу	Сумма штрафных санкций i-го заказа	Кол-во брака для i-го заказа	Стоимость ресурсов	Стоимость рабочей силы	Дополнительные затраты	Прямые затраты	Накладные расходы	Прибыль i-го заказа	Отказ потребителя
i	k_i	N_i	T_i	p_i	S_i	$x_i + y_i$	y_i	h_i	b_i	s_i рес	s_i *раб*	$^*y_i$	$S_{i\,\text{прям}}$	$S_{i\,\text{накл}}$	Прі	$r_j = 3$
1	8	32	1	8	336,00	32	0	0	0	33,60	67,20	0,00	100,80	168,00	67,20	0
2	15	60	4	15	418,50	15	0	0	0	41,85	83,70	0,00	125,55	209,25	83,70	0
3	41	164	4	41	1107,00	41	6	0	1	110,70	221,40	3,00	335,10	553,50	218,40	0
4	20	80	4	20	540,00	20	5	0	0	54,00	108,00	2,50	164,50	270,00	105,50	0
5	17	68	2	17	569,16	34	0	0,00	0	56,92	113,83	0,00	170,75	284,58	113,83	0
6	55	220	4	55	0,00	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00	1
7	15	60	4	15	418,50	15	14	0	3	41,85	83,70	7,00	132,55	209,25	76,70	0
8	12	48	2	12	401,76	24	24	0	2	40,18	80,35	12,00	132,53	200,88	68,35	0
9	15	60	2	15	502,20	30	0	0	1	50,22	100,44	0,00	150,66	251,10	100,44	0
10	18	72	2	18	602,64	36	17	0	0	60,26	120,53	8,50	189,29	301,32	112,03	0
11	5	20	4	5	135,00	5	0	13,5	4	13,50	27,00	0,00	40,50	67,50	13,50	0
12	25	100	4	25	675,00	25	0	67,5	20	67,50	135,00	0,00	202,50	337,50	67,50	0
13	22	88	2	22	712,80	44	24	0	0	71,28	142,56	12,00	225,84	356,40	130,56	0
14	52	208	4	52	1326,00	52	2	0	3	132,60	265,20	1,00	398,80	663,00	264,20	0
15	38	152	4	38	1026,00	38	0	0	2	102,60	205,20	0,00	307,80	513,00	205,20	0
16	18	72	2	18	602,64	36	36	0	0	60,26	120,53	18,00	198,79	301,32	102,53	0
17	15	60	4	15	405,00	15	15	0	1	40,50	81,00	7,50	129,00	202,50	73,50	0
18	17	68	2	17	569,16	34	22	0	0	56,92	113,83	11,00	181,75	284,58	102,83	0
19	14	56	2	14	468,72	28	0	0	1	46,87	93,74	0,00	140,62	234,36	93,74	0
20	5	20	1	5	210,00	20	0	0	0	21,00	42,00	0,00	63,00	105,00	42,00	0
Итого	**427**	**1708**	**58**	**427**	**11026,08**	**544**	**165**	**81**	**38**	**1102,608**	**2205,216**	**82,5**	**3390,32**	**5513,04**	**2041,72**	**1**

Таблица 4.3

Временные показатели модели

			Предпроизводственный норматив μ								Время на изготовление продукции t						
Номер заказа	Кол-во заказанных партий	Время прихода заказа	Банковские дни	Время на заказ МТР	Время на проплату МТР	Время на доставку МТР	Время на входной контроль качества МТР	Время на подгот. констр.-технол. докум.	Нормативное время	Время на ожидание осв-я произв. мощн.	Время на производство продукции	Время на контроль качества готовой продукции	Время на исправление брака	Время на предпродажную подготовку и упаковку	Фактическое время	Кол-во дней просрочки	Время на доставку потребителю
i	k_i	Θ_i					T_i*0,05		μ	U_i	T_i*0,8	T_i*0,1	$0{,}7*b_i^*$ (T/N_i)	T_i*0,1	T_i^*	τ_i	
1	8	2	2	1	1	1	0,1	0	5	0	0,8	0,1	0	0,1	1	0	1
2	15	3	2	1	1	1	0,2	0	5	0	3,2	0,4	0	0,4	4	0	1
3	41	5	2	1	1	3	0,2	2	9,2	0	3,2	0,4	0	0,4	4	0	2
4	20	6	2	1	1	2	0,2	0	6,2	1	3,2	0,4	0	0,4	4	0	1
5	17	7	2	1	1	1	0,1	1	6,1	2	1,6	0,2	0	0,2	2	0	1
6	55	8	2	1	1	3	0,2	2	9,2	3	0	0	0	0	0	0	0
7	15	9	2	1	1	2	0,2	0	6,2	0	3,2	0,4	0,1	0,4	4	0	1
8	12	10	2	1	1	2	0,1	0	6,1	0	1,6	0,2	0,1	0,2	2	0	1
9	15	11	2	1	1	2	0,1	0	6,1	0	1,6	0,2	0	0,2	2	0	1
10	18	12	2	1	1	3	0,1	0	7,1	0	1,6	0,2	0	0,2	2	0	1
11	5	14	2	1	1	3	0,2	3	10,2	0	3,2	0,4	0,6	0,4	5	1	1
12	25	15	2	1	1	3	0,2	0	7,2	0	3,2	0,4	0,6	0,4	5	1	1
13	22	16	2	1	1	2	0,1	0	6,1	0	1,6	0,2	0	0,2	2	0	1
14	52	17	2	1	1	3	0,2	1	8,2	2	3,2	0,4	0	0,4	4	0	2
15	38	18	2	1	1	3	0,2	0	7,2	5	3,2	0,4	0	0,4	4	0	1
16	18	20	2	1	1	2	0,1	0	6,1	0	1,6	0,2	0	0,2	2	0	1
17	15	22	2	1	1	3	0,2	0	7,2	0	3,2	0,4	0	0,4	4	0	1
18	17	25	2	1	1	2	0,1	0	6,1	0	1,6	0,2	0	0,2	2	0	1
19	14	27	2	1	1	1	0,1	1	6,1	0	1,6	0,2	0	0,2	2	0	1
20	5	30	2	1	1	1	0,1	0	5,1	0	0,8	0,1	0	0,1	1	0	1

Таблица 4.4

Диаграмма Ганта «Используемые производственные мощности»

Номер зак *i* / дни	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			8																												
2				15	15	15	15																								
3						41	41	41	41																						
4								20	20	20	20																				
5										17	17																				
6	отказ Потребителя																														
7										15	15	15	15																		
8											12	12																			
9												15	15																		
10													18	18																	
11															5	5	5	5													
12																25	25	25	25												
13																	22	22													
14																				52	52	52	52								
15																								38	38	38	38				
16																					18	18									
17																							15	15	15	15					
18																										17	17				
19																												14	14		
20																															5

Условные обозначения:

– день выполнения заказа

– день ожидания освобождения производственных мощностей

– дополнительный день выполнения заказа, возникающий из-за предшествующего ожидания освобождения производственных мощностей

Таблица 4.5

Диаграмма Ганта «Задействованное число основных рабочих»

Номер зак i / дни	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1			32																												
2				15	15	15	15																								
3						35	35	35	35																						
4								15	15	15	15																				
5										34	34																				
6	отказ Потребителя																														
7										1	1	1	1																		
8											0	0																			
9												30	30																		
10													19	19																	
11															5	5	5	5													
12																25	25	25	25												
13																	20	20													
14																				50	50	50	50								
15																								38	38	38	38				
16																					0	0									
17																						0	0	0	0						
18																										12	12				
19																												28	28		
20																															20

4.2. Моделирование ЗАО «Уральская индустриальная группа»

ЗАО «Уральская индустриальная группа» (ЗАО «УИГ») занимается производством и продажей пластиковых окон. Окна производятся как на собственном производстве, так и по заказу ЗАО «УИГ» на заводе изготовителе СОК в г. Самаре, затем осуществляется доставка окон на объект и их монтаж. Основной процесс ЗАО «УИГ», ориентированный на клиента, изображен на рис. 4.3.

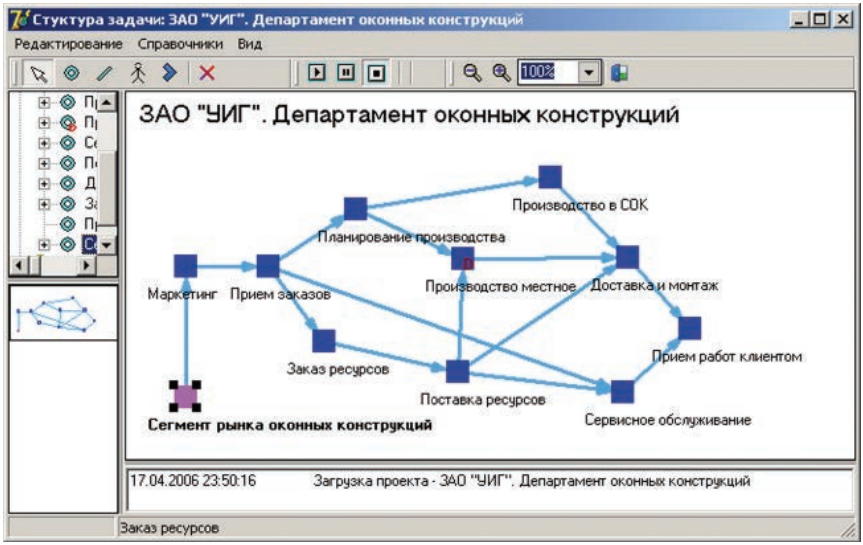


Рис. 4.3. Основной процесс ЗАО «УИГ»

Постановка задачи: построить модель и проанализировать положение предприятия на рынке услуг, оценить деятельность конкурентов и с учетом их поведения скоординировать собственную стратегию.

Цель: разработать алгоритм поведения и ценовую стратегию для увеличения доли рынка и перехода на новый технологический уровень, повышающий конкурентоспособность предприятия. Необходимо рассчитать долгосрочную ценовую стратегию, позволяющую на первом этапе – путем снижения цен – увеличить объем продаж (долю рынка) за счет вытеснения с рынка слабых конкурентов; на втором – путем постепенного повышения цены, не снижая доли рынка, получить дополнительную прибыль, позволяющую модернизировать

производственные мощности предприятия новым современным оборудованием, которое позволит повысить конкурентоспособность (увеличить качество и количество выпускаемой продукции) и привлечь новые группы потребителей на рынке. Определить оптимальную долю рынка (объем производства) и цену товара, позволяющие получить необходимую дополнительную прибыль.

Исходные данные:

– доля рынка 6,6 %;

– объем продаж $V = 2000$ м^2 в месяц;

– цена 1 м^2 –5500 руб.;

– оформление заявки в СОК (Самарский оконный завод) – 1 день;

– производство окон на заводе – изготовителе – 5 дней;

– доставка – 2 дня;

– конкурентная среда характеризуется следующими параметрами: доля рынка, количество конкурентов на рынке, интенсивность борьбы, цены конкурентов, реакция по времени и цене, оценка конкурентоспособности;

– спрос характеризуется следующими параметрами: эластичность по цене, сезонность, емкость рынка.

Процессы предприятия, которые рассматриваются в модели: процесс производства; процесс продаж; процесс монтажа; процесс сервисного обслуживания. В результате проведенного системного анализа основные процессы деятельности ЗАО «УИГ» были описаны в стандарте IDEF0 и собраны данные, необходимые для моделирования.

4.2.1. Процесс производства

Процесс производства включает в себя производство оконных и дверных конструкций. Каждый процесс производства состоит из определенного количества операций. Определены ресурсы, механизмы и специалисты, которые участвуют в процессе.

На каждую технологическую операцию требуется не менее одного специалиста. Последовательность операций в процессе производства окон и дверей одинакова, но в отдельных технологических операциях используется разное оборудование. Диаграмма декомпозиции процесса «Производство местное» представлена на рис. 4.4.

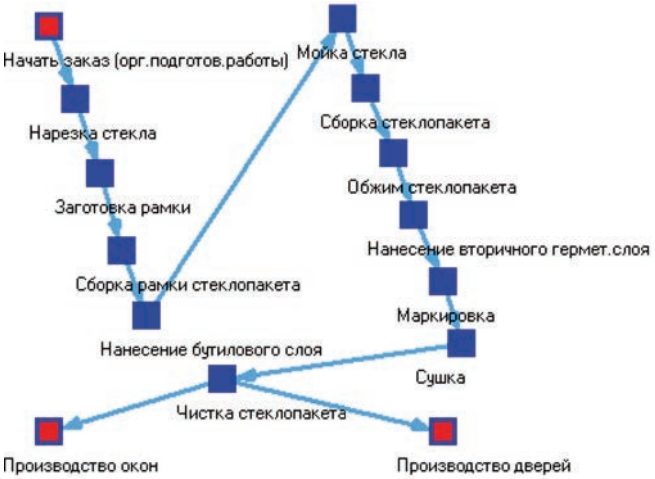


Рис. 4.4. Процесс производства

В производственном процессе участвует 30 человек и 26 видов различного оборудования. Диаграмма декомпозиции процессов «Производство стеклопакетов» и «Производство окон» изображена на рис. 4.5 и 4.6 соответственно.

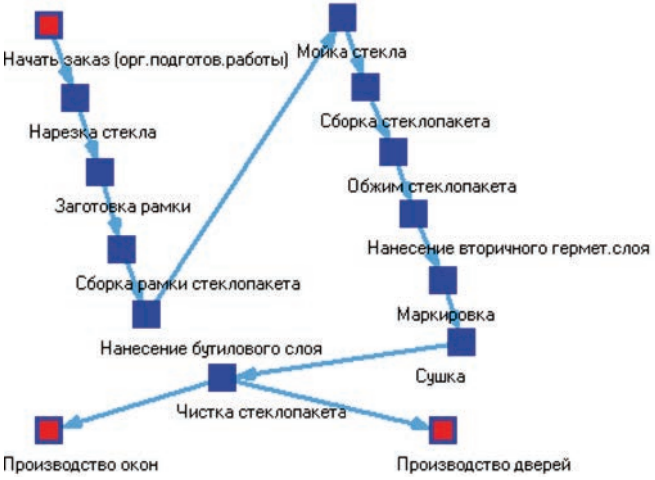


Рис. 4.5. Производство стеклопакетов

Рис. 4.6. Производство окон

Диаграмма декомпозиции процесса «Участок дополнительных элементов» представлена на рис. 4.7.

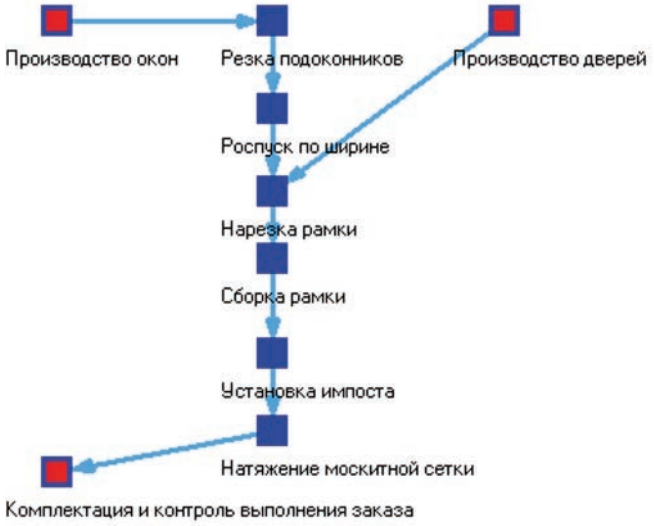


Рис. 4.7. Участок дополнительных элементов

4.2.2. Процесс монтажа

Монтаж включает в себя доставку и установку готовых конструкций. Число монтажников – 30 человек (15 звеньев, звено – 2 чел). Площадь одного окна в среднем 2,5 м^2. Одно звено в среднем за час работы выполняет установку 1,25 м^2 окон.

4.2.3. Модели агентов

Перед построением моделей агентов были описаны основные цели ЗАО «УИГ» и представлены в системе сбалансированных показателей (рис. 4.8).

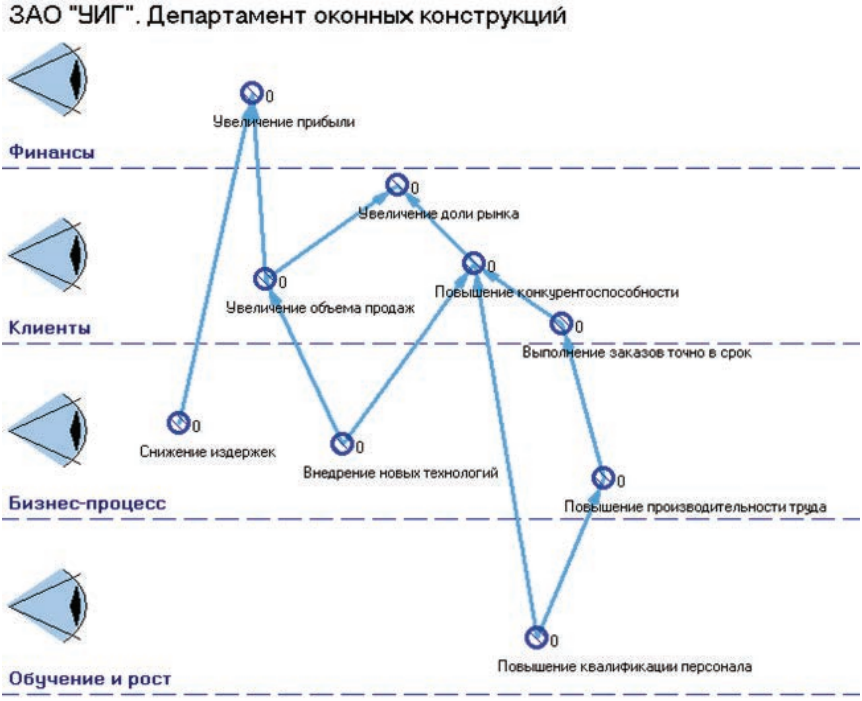


Рис. 4.8. Общая диаграмма BSC ЗАО «УИГ»

Ниже приводится описание моделей агентов ЗАО «УИГ».

Таблица 4.6

База знаний Агента 1А: «Потребитель»

(Цель Агента 1А: $S_i \rightarrow \min S_i\{n,k,t,s\}$, т.е. минимизировать стоимость заказа при неизменном качестве)

№ п/п	Если	То	Описание
1	Начало моделирования Θ = 1 и пока Θ < = 365	(Генерируем запросы с частотой (С85 / 30) в день, количество продукции *N* от1 до 110 со средним значением 5,3, время выполнения Т от 0,1 до 8 со средним значением 2)(0) ∧ (определяем значение номера офиса *k*, *k* от 1 до 7, вероятность выбора номера *k* = С*k*5 / С85)(0) ∧ (начальнику отдела продаж 2А сообщение о контакте с офисом *k*)(0)	Потребитель 1А отправляет в офис*k* запрос (контакт) о заказе продукции *N*, срок выполнения *T* первоначально частота генерации (С85 / 30) запросов в день (единица модельного времени) при выборе *N* и Т используется треугольное распределение вероятности (можно заменить на нормальное распределение). Номер офиса *k* выбирается с вероятностью по матрице С*ij* 5-й столбец, каждый раз выбирается только один номер. Высылаем сгенерированные запросы 2А.
2	От начальника отдела продаж 2А получено сообщение о подписании договора по заказу *Zi* в офисе *k*, стоимость заказа S_i	(перечисление денежных средств *Si* по договору) (3)	Считаем, что заказу *Zi* сопутствует вся информация по заказу: *Ni, Xi, Ti, Si, Pi, k* – номер офиса и т. д.
3	(от начальника отдела продаж 2А получено сообщение, что заказ *Zi* выполнен) ∧ ∧ ($T_i^* \le T_i$)	(принять заказ) (0)	
4	(от начальника отдела продаж 2А получено сообщение, что заказ *Zi* выполнен) ∧ ∧ ($T_i^* > T_i$)	(принять заказ) (0) ∧ (получить штрафные санкции H_i) (3)	

Таблица 4.7

База знаний Агента 2А: «Начальник отдела продаж»
(Цель Агента 2А: Максимизация прибыли при неизменной прямой себестоимости)

№ п/п	Если	То	Описание
1	2	3	4
1	От Конкурента 5А сообщение, что *m*-й конкурент изменил цену	(запомним действующую цену Ц = E10 7) (E10 5) ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ (D цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5)	После изменения цены конкурентом *m* начальник отдела продаж 2А вызывает процедуру proc определения цены для корректировки своей цены через время реакции, описанное в матрице E10 5 (10-я строка, 5-й столбец). Расчет D цены – изменение цены в процентах и передача полученного значения в процедуру реакции на изменение цены
2	(вызвана процедура Proc2 реакции на изменение цены) ∧ ∧ (–10 % < *D* цены < = < = 10 %)	(Ц = E10 7)(0) ∧ (D KK = –3 * D цены)(0) ∧ (C*k*5 = C*k*5*(100 % + D KK) для всех *k*-офисов)(0) ∧ (D КД/KK = D цены)(0) ∧ ∧ C*k*9 = C*k*9*(100 % + D КД/KK) для всех *k*-офисов)(0) ∧ (DКД = –(C14- C15 * C19) – – (C24 – C25 * C29)– ...–(CN4- CN5 * CN9) где *N* – количество офисов) (0) ∧ ∧ (C*k*4 = C*k*5 * C*k*9 для всех *k*-офисов)(0) ∧ ∧ (DV = DКД * sred)(0) ∧ (E10 2 = = E10 2 + DV)(0) ∧ (E1 2 = E1 2-DV* E1 1,..., E9 2 = E9 2 – DV* E9 1,E11 2 = E11 2 – DV* E11 1 для всех *m* конку-рентов)(0)∧ (E12 2 = E1 2 + + E2 2 + ... + E11 2)(0) ∧ (E1 1 = E1 2/E 12 2, E2 1 = = E2 2/E12 2,..., E11 1 = E11 2/E12 2)(0)	Запоминаем новое значение цены. Изменяем значения C*k*5 – KK в месяц по всем офисам в соответствии с выведенным линейным законом. Изменяем значения C*k*9 – коэффициента КД/KK в месяц по всем офисам в соответствии с выведенным линейным законом. Расчет суммарного изменения количества договоров DКД в штуках по всем офисам. Расчет изменения объемов продаж. Перераспределение изменения объемов продаж между конкурентами в соответствии с долей рынка Пересчет доли рынка E12 2 – весь рынок в м^2 Em1 – доля рынка, %

Продолжение табл. 4.7

1	2	3	4
3	(вызвана процедура Proc2 реакции на изменение цены) ∧ ∧ (10 % < *D*цены < = 20 %)	(Ц = E10 7)(0) ∧ (D KK = – Dцены-20 %) (0) ∧ ∧ (C*k*5 = C*k*5*(100 % + D KK) для всех *k*-офисов) (0) ∧ (D КД/KK = 0,5 * Dцены + 5 %)(0) ∧ ∧ C*k*9 = C*k*9*(100 % + D КД/KK) для всех *k* – офисов)(0) ∧ (DКД = –(C14 – C15 * C19) – – (C24 – C25 * C29)– ...–(CN4 – CN5 * CN9), где *N* – количество офисов) (0) ∧ (C*k*4 = C*k*5 * *C*k*9 для всех *k*-офисов)(0) ∧ (DV = DКД * sred)(0) ∧ ∧ (E10 2 = E10 2 + DV)(0) ∧ (E1 2 = = E1 2 – DV* E1 1,..., E9 2 = E9 2-DV* E9 1, E11 2 = E11 2 – DV* E11 1 для всех m конкурентов)(0)∧(E12 2 = E1 2 + E2 2 + ... + + E11 2)(0) ∧ (E1 1 = E1 2/E 12 2, E2 1 = = E2 2/E12 2,..., E11 1 = E11 2/E12 2)(0)	Запоминаем новое значение цены. Изменяем значения Ck5 – KK в месяц по всем офисам в соответствии с выведенным линейным законом. Изменяем значения Ck9 – коэффициента КД/KK в месяц по всем офисам в соответствии с выведенным линейным законом. Расчет суммарного изменения количества договоров DKД в штуках по всем офисам. Расчет изменения объемов продаж. Перераспределение изменения объемов продаж между конкурентами в соответствии с долей рынка. Пересчет доли рынка E12 2 – весь рынок в м^2 Em1 – доля рынка, %
4	(вызвана процедура Proc2 реакции на изменение цены) ∧ ∧ (–20 % < *D*цены <= –10 %)	(Ц = E10 7)(0) ∧ (D KK = -2 * Dцены + 10 %) (0) ∧ ∧ (C*k*5 = C*k*5*(100 % + D KK) для всех *k* – офисов)(0) ∧ (D КД/KK = 0,2 * Dцены – – 8 %)(0) ∧ C*k*9 = C*k*9*(100 % + D КД/KK) для всех *k* – офисов)(0) ∧ (DКД = – (C14 – C15 * *C19) – (C24 – C25 * C29)– ...–(CN4– CN5 * CN9) где N– количество офисов) (0) ∧ (C*k*4 = = C*k*5 * C*k*9 для всех *k*-офисов)(0) ∧ (DV = = DКД * sred)(0) ∧(E10 2 = E10 2 + DV)(0) ∧ ∧ (E1 2 = E1 2 – DV* E1 1,..., E9 2 = E9 2 – DV* E9 1, E11 2 = E11 2-DV* E11 1 для всех *m* конкурентов)(0)∧(E12 2 = E1 2 + E2 2 + ... + + E11 2)(0) ∧ (E1 1 = E1 2/E 12 2, E2 1 = = E2 2/E12 2,..., E11 1 = E11 2/E12 2)(0)	Запоминаем новое значение цены. Изменяем значения Ck5 – KK в месяц по всем офисам в соответствии с выведенным линейным законом Изменяем значения Ck9 – коэффициента КД/KK в месяц по всем офисам в соответствии с выведенным линейным законом. Расчет суммарного изменения количества договоров DKД в штуках по всем офисам. Расчет изменения объемов продаж. Перераспределение изменения объемов продаж между конкурентами в соответствии с долей рынка Пересчет доли рынка E12 2 – весь рынок в м^2 Em1 – доля рынка, %

1	2	3	4
5	(вызвана процедура Proc2 реакции на изменение цены) ∧ ∧ (20 % < *D*цены <= 30 %)	(Ц = E10 7)(0) ∧ (D KK = – Dцены-20 %) (0) ∧ ∧ (C*k*5 = Ck5*(100 % + D KK) для всех *k*-офисов)(0) ∧ ∧ (D КД/KK = 0,2 * Dцены + 11 %)(0) ∧ ∧ Ck9 = Ck9*(100 % + D КД/KK) для всех *k* – офисов)(0) ∧ (DКД = -(C14 – C15 * C19) – – (C24 – C25 * C29) – ... – (CN4 – CN5 * CN9) где *N*– количество офисов) (0) ∧ (C*k*4 = = C*k*5 * C*k*9 для всех *k*-офисов)(0) ∧ (DV = = DКД * sred)(0) ∧(E10 2 = E10 2 + DV)(0) ∧ ∧ (E1 2 = E1 2-DV* E1 1,..., E9 2 = E9 2-DV* E9 1, E11 2 = E11 2 – DV* E11 1 для всех *m* конкурентов)(0) ∧ (E12 2 = E1 2 + E2 2 + ... + + E11 2)(0) ∧(E1 1 = E1 2/E 12 2, E2 1 = = E2 2/E12 2,..., E11 1 = E11 2/E12 2)(0)	Запоминаем новое значение цены Изменяем значения C*k*5 – KK в месяц по всем офисам в соответствии с выведенным линейным законом Изменяем значения C*k*9 – коэффициента КД/KK в месяц по всем офисам в соответствии с выведенным линейным законом. Расчет суммарного изменения количества договоров DКД в штуках по всем офисам. Расчет изменения объемов продаж. Перераспределение изменения объемов продаж между конкурентами в соответствии с долей рынка Пересчет доли рынка E12 2 – весь рынок в м^2 Em1 – доля рынка, %
6	(вызвана процедура Proc2 реакции на изменение цены) ∧ ∧ (–30 % < *D*цены <= –20 %)	(Ц = E10 7)(0) ∧ (D KK = –5 * Dцены – 50 %) (0) ∧ ∧ (C*k*5 = C*k*5*(100 % + D KK) для всех *k* – офисов)(0) ∧ ∧ (D КД/KK = 0,3 * Dцены – 6 %) (0) ∧C*k*9 = C*k*9*(100 % + D КД/KK) для всех k-офисов)(0) ∧ (DКД = -(C14 – C15 * C19) – – (C24 – C25 * C29) – ... – (CN4 – CN5 * CN9) где N – количество офисов) (0) ∧ (C*k*4 = C*k*5 * *C*k*9 для всех *k*-офисов)(0) ∧ (DV = DКД * * sred)(0) ∧(E10 2 = E10 2 + DV)(0) ∧ (E1 2 = = E1 2-DV* E1 1,..., E9 2 = E9 2-DV* E9 1, E11 2 = E11 2-DV* E11 1 для всех *m* конкурентов)(0)∧(E12 2 = E1 2 + E2 2 + ... + + E11 2)(0) ∧ (E1 1 = E1 2/E 12 2, E2 1 = = E2 2/E12 2,..., E11 1 = E11 2/E12 2)(0)	Запоминаем новое значение цены. Изменяем значения C*k*5 – KK в месяц по всем офисам в соответствии с выведенным линейным законом Изменяем значения C*k*9 – коэффициента КД/KK в месяц по всем офисам в соответствии с выведенным линейным законом. Расчет суммарного изменения количества договоров DКД в штуках по всем офисам. Расчет изменения объемов продаж. Перераспределение изменения объемов продаж между конкурентами в соответствии с долей рынка. Пересчет доли рынка E12 2 – весь рынок в м^2 Em1 – доля рынка, %

Продолжение табл. 4.7

1	2	3	4
7	от потребителя 1А сообщение о контакте с офисом *k* о заказе продукции *N*, срок выполнения *T*	(Ск 15 = Ск 15 + 1)(0) ∧ (с вероятностью Random((Ск9) * f(Θ)) Ск 14 = Ск 14 + 1 т.е. договор подписан)(0)	Увеличить счетчик фактических контактов *k*-го офиса на 1 (Ск 15). С вероятностью (Ск9 * f(Θ)) подписывается договор, тогда счетчик фактических договоров увеличивается на 1 (Ск 14). f(Θ) – коэффициент сезонности берется из D1j в соответствии с условием D6j на день года
8	подписан договор с потребителем 1А в офисе *k*	(*i* = *i* + 1) (0) ∧ (открыть заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i)(0) ∧ (Расчет стоимости *Si* в соответствии с матрицей Fij)(0) ∧ (Расчет количества звеньев рабочих *Xi* в соответствии с матрицей Gij)(0) ∧ (расчет количества мощностей *Pi* = *Xi* * *p*)(0) ∧ (стоимость заказа *Si*сообщить потребителю 1А)(0)	Увеличить общий счетчик заказов. Срок выполнения рассчитывается только на производство и монтаж. Количество звеньев рабочих надо округлить до большего целого (звено – 2 чел)
9	от потребителя 1А по договору *k*-го офиса поступили денежные средства *Si* по заказу *Zi*	(*Vk* = *Vk* + *Si*)(0) ∧ (*Qk* = *Qk* + *Ni*)(0) ∧ ∧ (начальнику производственного участка 4А сообщение о заказе *Zi* (*Ni, Xi, Pi, Ti*))(0)	Увеличили объем продаж *k*-го офиса в м^2 и в тыс. руб.
10	(от начальника производ-ственного участка 4А получено сообщение, что произведено по заказу *Zi* продукции N_i за время T_i*) ∧ ($T^*_i <= T_i$)	(потребителю 1А сообщение о выполнении заказа) (0)	

1	2	3	4
11	(от начальника производственного участка 4А получено сообщение, что произведено по заказу *Zi* продукции N_i за время T_i*) ∧ ($T^*_i > T_i$)	($H_i = 0{,}1\,\%\ (T^*_i - T_i)S_i$) (0) ∧ (*Vk* = *Vk* – *Hi*) (0) ∧ (потребителю 1А сообщение о выполнении заказа и перечислении штрафа)(3)	Рассчитать сумму штрафных санкций
12	(Θ = 32) ∨ (Θ = 60) ∨ ∨ (Θ = 91) ∨(Θ = 121) ∨ ∨ (Θ = 152) ∨ (Θ = 182) ∨ ∨ (Θ = 213) ∨ (Θ = 244) ∨ ∨ (Θ = 274) ∨ (Θ = 305) ∨ ∨ (Θ = 335) ∨ (Θ = 366)	(НРк = ПЗ * Ск8 + МБ/ С88 * Ск8 + СодО) ∧ ∧ (Пк = *Vk*– С***Qk* – НР*k*) ∧ (Э*k* = V*k* * *P* /НР*k*) ∧ ∧ (*kf* = max(Э*k*), *kd* = max(Э*k* кроме *kf*), НР*kf* = НР*kf* + Прем* С*kf*8, НР*kd* = НР*kd* + Прем* С*kd*8, П*kf* = П*kf*-Прем* С*kf*8, П*kd* = П*kd*-Прем* С*kd*8, Эк*f* = V*kf* * Р /НРк*f*, Эк*d* = *Vkd* * Р /НРк*d*) ∧ (П = Сумм(Пk), *V* = Сумм(*Vk*), *Q* = Сумм(*Qk*))(0) ∧ (вызов процедуры Proc3 анализа отчетного месяца)(0)	Θ – текущий день (единица модельного времени) Закончился отчетный период – месяц. Для каждого *k*-го офиса считаем НР*k* – накладные расходы, П*k*-прибыль до налогообложения, Э*k* – эффективность офиса, находим два самых эффективных офиса, даем премию работникам, пересчитываем накладные расходы, прибыль и эффективность
13	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (Θ ≠ 60) ∧ (Θ ≠ 121)) ∧ ∧ (Θ ≠ 244)	(E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры proc определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ (D цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Вызываем процедуру Proc определения цены для корректировки своей цены через время реакции, описанное в матрице E10 5 (10-я строка, 5-й столбец). Расчет D цены – изменения цены в процентах и передача полученного значения в процедуру реакции на изменение цены

Продолжение табл. 4.7

1	2	3	4
14	(процедура Proc3 анализа отчетного месяца)∧ ∧ (Θ = 60) ∧ ((*Q* < E10 2) ∧ ∧ (П > = 0,25**V*))	(Цикл по всем *j*от 1 до 8 (если B*j*2 < 0,1 то B92 = B92 + Bj2, Bj2 = 0))(0) ∧ (Цикл по всем *k* C*k*9 = 110 % * C*k*9)(0) ∧ (E10 2 = *Q*)(0) ∧ (запомним действующую цену Ц = E10 7)(E105) ∧ ∧ (вызов процедуры Proc1 определения цены) (E10 5) ∧ (Сообщение Конкуренту 5A, что начальник отдела продаж 2A изменяет цену)(10 5) ∧ ∧ (D цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 2 мес. доля рынка уменьшилась, меняем инструменты маркетинга, убираем все мелкие инструменты и мотивируем персонал, эффективность возрастает на 10 %. Вызываем процедуру proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменения цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены
15	(процедура Proc3 анализа отчетного месяца)∧ ∧ (Θ = 121) ∧ ((Q < E10 2) ∧ ∧ (П > = 0,25**V*))	(Стратегия – проводить тренинги) ∧ (E10 2 = = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5A, что начальник отдела продаж 2A изменяет цену)(E10 5) ∧ (*D*цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 4 мес доля рынка уменьшилась, проводим тренинги. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменения цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены
16	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (Θ = 244) ∧ ((Q < E10 2) ∧ ∧ (П > = 0,25**V*))	(B32 = B32 + B92, B92 = 0)(0) ∧ (C*k*9 = = 105 % * C*k*9)(0) ∧ (E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5A, что начальник отдела продаж 2A изменяет цену)(E10 5) ∧ (D цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 8 мес. доля рынка уменьшилась, меняем инструменты маркетинга, убираем мотивацию персонала, вкладываем в рекламу на телевидении, эффективность возрастает на 5 %. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменения цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены

Продолжение табл. 4.7

1	2	3	4
17	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (Θ = 60) ∧ ((*Q* < E10 2) ∧ ∧ (П < 0,25**И*))	(Цикл по всем *j* от 1 до 8 (если B*j*2 < 0,1 то B92 = B92 + Bj2, Bj2 = 0))(0) ∧ ∧ (Цикл по всем *k* C*k*9 = 110 % * C*k*9)(0) ∧ ∧ (МБ = МБ*(100 % + 10 %)(0) ∧ ∧ (E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ ∧ (D цены = (E10 7 – Ц) / Ц – вызов процедуры Proc2 реакции на изменение цены) (E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 2 мес. доля рынка уменьшилась, меняем инструменты маркетинга, убираем все мелкие инструменты и мотивируем персонал, эффективность возрастает на 10 %. Закончился отчетный период – месяц, и доля рынка стала меньше предыдущей и прибыль снизилась – стратегия на увеличение бюджета МБ. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменение цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены
18	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (Θ = 121) ∧ ((*Q* < E10 2) ∧ ∧ (П < 0,25**И*))	(Стратегия – проводить тренинги) ∧ (МБ = = МБ*(100 % + 10 %)(0) ∧ (E10 2 = *Q*)(0) ∧ ∧ (запомним действующую цену Ц = E10 7) (E10 5) ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ (D цены = = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 4 мес. доля рынка уменьшилась, проводим тренинги. Закончился отчетный период – месяц, доля рынка стала меньше предыдущей и прибыль снизилась – стратегия на увеличение бюджета МБ. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменения цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены

Продолжение табл. 4.7

1	2	3	4
19	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (Θ = 244) ∧ ((*Q* < E10 2) ∧ ∧ (П < 0,25**V*))	(B32 = B32 + B92, B92 = 0)(0) ∧ (C*k*9 = = 105 % * C*k*9)(0) ∧ (МБ = МБ*(100 % + 10 %)(0) ∧ ∧ (E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ ∧ (Dцены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены) (E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Через 8 мес. доля рынка уменьшилась, меняем инструменты маркетинга, убираем мотивацию персонала, вкладываем в рекламу на телевидении, эффективность возрастает на 5 %. Закончился отчетный период – месяц, доля рынка стала меньше предыдущей и прибыль снизилась – стратегия на увеличение бюджета МБ. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменение цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены
20	(процедура Proc3 анализа отчетного месяца) ∧ (((Θ = 60)∧(*Q* > E10 2)) ∨ ∨ ((Θ = 121) ∧ (*Q* > E10 2) ∧ (П < 0,25**V*)) ∨ ((Θ = 244) ∧ ∧ (*Q* > E10 2) ∧ (П < 0,25**V*)))	(Цикл по всем *k*. C*k*9 = 90 % * C*k*9)(0) ∧ ∧ (МБ = МБ*(100 % – 10 %)(0) ∧ (E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры Proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ (D цены = = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены)(E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Закончился отчетный период – месяц, и доля рынка стала больше предыдущей, прибыль достаточная- стратегияна уменьшение бюджета МБ; эффективность уменьшается на 10 %. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменение цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены

Продолжение табл. 4.7

1	2	3	4
21	(процедура Proc3 анализа отчетного месяца) ∧ ∧ ((Θ = 121) ∨ (Θ = 244)) ∧ ∧ ((Q > E10 2) ∧ (П > 0,25*V))	(Цикл по всем *k* C*k*9 = 90 % * C*k*9)(0) ∧ (МБ = = МБ*(100 % –10 %)(0) ∧ (Открыть *k*-ый офис в новом районе КК*k* = КК*k**(100 % – 35 % / (*N* – 1)) где (N – 1) – количество офисов C*kN* = 2 НР = НР + 100 П = П – 100)(0) ∧ ∧ (E10 2 = Q)(0) ∧ (запомним действующую цену Ц = E10 7)(E10 5) ∧ (вызов процедуры proc1 определения цены)(E10 5) ∧ (Сообщение Конкуренту 5А, что начальник отдела продаж 2А изменяет цену)(E10 5) ∧ ∧ (D цены = (E10 7 – Ц) / Ц вызов процедуры Proc2 реакции на изменение цены) (E10 5) ∧ (вызов процедуры Proc4 закрытия отчетного месяца) (0)	Закончился отчетный период – месяц, и доля рынка стала больше предыдущей, прибыль достаточная- стратегия на уменьшение бюджета МБ эффективность уменьшается на 10 %. Открыть *k*-й офис в новом районе. Вызываем процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрицеE10 5 (10-я строка, 5-й столбец). Расчет D цены – изменение цены в процентах и передача полученного значения в процедуру Proc2 реакции на изменение цены
22	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (МБ > 110 % * (*V* * *R* / 15))	(МБ = МБ*(100 % – 10 %))(0)	Если маркетинговый бюджет превышает маржинальный доход более чем на 10 %, то надо уменьшить МБ (15 – рублей отдачи с рубля, вложенного в рекламу)
23	(процедура Proc3 анализа отчетного месяца) ∧ ∧ (МБ < 90 % * (*V* **R*/ 15))	(МБ = МБ*(100 % + 10 %)(0)	Если маркетинговый бюджет меньше маржинального дохода более чем на 10 %, то надо увеличить МБ
24	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 32)	(V1*k* = V*k*) ∧ (Q1*k* = Q*k*) ∧ (П1*k* = П*k*) ∧ ∧ (Э1*k* = Э*k*) ∧ (V*k* = 0) ∧ (Q*k* = 0) ∧ ∧ (КК*k* = 0) ∧ (КД*k* = 0)	Закончился отчетный период – январь месяц Запомнили объем продаж ($м^2$ и тыс. руб.), прибыль и эффективность по каждому офису. Потом разработаем блок анализа по году (по результатам моделирования)

Продолжение табл. 4.7

1	2	3	4
25	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 60)	(V2k = Vk) ∧ (Q2k = Qk) ∧ (П2k = Пk) ∧ ∧ (Э2k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – февраль месяц
26	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 91)	(V3k = Vk) ∧ (Q3k = Qk) ∧ (П3k = Пk) ∧ ∧ (Э3k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – март месяц
27	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 121)	(V4k = Vk) ∧ (Q4k = Qk) ∧ (П4k = Пk) ∧ ∧ (Э4k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – апрель месяц
28	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 152)	(V5k = Vk) ∧ (Q5k = Qk) ∧ (П5k = Пk) ∧ ∧ (Э5k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – май месяц
29	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 182)	(V6k = Vk)∧ (Q6k = Qk) ∧ (П6k = Пk) ∧ ∧ (Э6k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – июнь месяц
30	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 213)	(V7k = Vk) ∧ (Q7k = Qk) ∧ (П7k = Пk) ∧ ∧ (Э7k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – июль месяц
31	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 244)	(V8k = Vk) ∧ (Q8k = Qk) ∧ (П8k = Пl) ∧ ∧ (Э8k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – август месяц
32	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 274)	(V9k = Vk) ∧ (Q9k = Qk) ∧ (П9k = Пk) ∧ ∧ (Э9k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – сентябрь месяц
33	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 305)	(V10k = Vk) ∧ (Q10k = Qk) ∧ (П10k = Пk)∧ ∧ (Э10k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – октябрь месяц

Окончание табл. 4.7

1	2	3	4
34	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 335)	(V11k = Vk) ∧ (Q11k = Qk) ∧ (П11k = Пk) ∧ ∧ (Э11k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0)	Закончился отчетный период – ноябрь месяц
35	(процедура Proc4 закрытия отчетного месяца) ∧ ∧ (Θ = 366)	(V12k = Vk) ∧ (Q12k = Qk) ∧ (П12k = Пk) ∧ ∧ (Э12k = Эk) ∧ (Vk = 0) ∧ (Qk = 0) ∧ ∧ (ККk = 0) ∧ (КДk = 0) Анализ V1...V12, Q1... Q12, П1.....П12, Э1...Э12	Закончился отчетный период – декабрь месяц. Закончилось время моделирования – 365 дней. Анализ итогов года, построение графиков по *V, Q*, П, Э
36	выработана стратегия – проводить тренинги	Проводятся тренинги в течении 1 месяца, начало t = Θ (МБ = МБ*(100 % – 10 %))(0)∧ ∧ (Сk9 = Сk9*105 %) (0) ∧ (МБ = = МБ* (100 % + 10 %)(30)	Тренинги проводятся в течение месяца за счет 10 % МБ, дадут увеличение КД/КК на 5 %, через месяц необходимо восстановить бюджет маркетинга
37	Офис продаж убыточен > = 2 месяца Сk1*(30 % + 35 %) <= Сk12	Закрытие k-го офиса, приводит к: НР = НР – Ск13 ККk = ККk*(100 % + 35 % / (N – 1)) где N – количество офисов до закрытия. Сk1 = 0 ..... Ck13 = 0)(0)	Если офис продаж в течение 2-х месяцев убыточен, то закрываем данный офис. Число контактов в других офисах возрастет на 35 % Увольняем менеджеров, обнуляем все параметры k-го офиса
38	((Θ = 32) ∨ (Θ = 60) ∨ ∨ (Θ = 91) v (Θ = 121) ∨ ∨ (Θ = 152) ∨ (Θ = 182) ∨ ∨ (Θ = 213) ∨ (Θ = 244) ∨ ∨ (Θ = 274) ∨ (Θ = 305) ∨ ∨ (Θ = 335) ∨ (Θ = 366)) ∧ ∧ (Q < E10 2) ∧ ∧ (закрылиk-йофис)	(Открыть k-й офис в другом районе ККk = ККk*(100 %-35 % / (N – 1)) где N – количество офисов до закрытия. С$k$$N$ = 2 НР = НР + 100 П = П – 100)(0)	Если доля рынка уменьшилась и закрыли офис, то открываем в другом районе
39	(КД/КК)i = = min(КД/КК)i ≥ 2 мес. и V_продаж ≥ V_продаж min	Перевести одного менеджера из этого офиса на стажировку в лучший офис, где (КД/КК) i = = max(КД/КК)i	i = 1, NN – число офисов. V_продаж min – минимально-допустимый объем продаж

Таблица 4.8

База знаний Агента 3А: «Начальник производственного участка»

Цель Агента 3А: Выполнение заказов точно в срок, повышение производительности труда

№ п/п	Если	*То*	Описание
1	2	3	4
1	От начальника отдела продаж 2А получено сообщение о начале выполнения заказа *Zi*	(заказ ресурсов у поставщиков в расчете на k_i партий) (0) ∧ (доставка ресурсов от поставщиков) (3) ∧ (входной контроль ресурсов) (1) ∧ (вызов процедуры Proc5 планирования выполнения заказа)(0)	Считаем, что заказу *Zi* сопутствует вся информация: *Ni, Xi, Ti, Si, Pi, Ki* и т.д. Считаем, что прямые затраты (материалы, зарплата рабочих, коммунальные платежи участка, транспорт участка и т.д.) учтены в прямой себестоимости продукции. Считаем, что на складе материалов есть некоторый запас, позволяющий начинать производств, не дожидаясь поступления материалов и комплектующих.
2	(ресурсы от поставщика не прошли входной контроль)	(возврат ресурсов поставщику) (1) ∧ ∧ (заказ недостающих ресурсов у поставщиков) (1)	Событие должно генерироваться с некоторой заданной вероятностью, например 0,5%. Возникновение события влечет потерю времени при малых запасах. При достаточных запасах на складах потери времени не возникает
3	(процедура Proc5 планирования) ∧ ∧ ($p^* \geq p_i$, т.е. есть свободные производственные мощности)	($\Theta_i = \Theta$ начинается выполнение заказа z_i) (0) ∧ ∧ ($p^* = p^* - p_i$) (0) ∧($x^* = x^* - x_i$) (0) ∧ ∧ (введение информации по загрузке производственных мощностей в массив плановой загрузки B: B[Θ] = B[Θ] + *Pi*, *B*[Θ + 1] = *B*[Θ + 1] + *Pi* ... *B*[Θ + *Ti*] = = *B*[Θ + *Ti*] + *Pi*)(0) ∧ (начальнику монтажного участка 4А сообщение о заказе *Zi* (*Ni, Xi, Pi, Ti*)(0)	p^* – свободные производственные мощности, $p^* = ST - \sum_{j=1}^{i} p_j$ (перначально $p^* = ST$), здесь j – бегущий индекс по идущим заказам x^* – число свободных основных рабочих, $x^* = PR - \sum_{j=1}^{i} x_j$ (первоначально $x^* = PR$)

Продолжение табл. 4.8

1	2	3	4
4	(процедура Proc5 планирования) ∧ ∧ ($p^* < p_i$, т.е. нет свободных производственных мощностей)	(заказ z_i. приостанавливается из-за нехватки производственных мощностей на 1 день)(1)	
5	(от начальника монтажного участка 4А сообщение о выполнении заказа Zi (Ni, Xi, Pi, Ti)(0)	(контроль качества продукции N_i) (0) ∧ (генерация количества брака b_i = [Random(0,5 % * Ni)])(0) ($T^*_i = \Theta - \Theta_i$) (0) ∧ ($x^* = x^* + x_i$) (0) ∧ ∧ ($p^* = p^* + p_i$) (0)	Событие должно генерироваться с некоторой заданной вероятностью, например 0,5 %
6	($b_i \neq 0$, т.е. часть продукции забракована)	(выделение рабочих на исправление брака в количестве [$Xi * b_{i/} Ni$] + 1) ∧ ∧ (выделение мощностей на исправление брака в количестве [$Pi * b_i / Ni$] + 1) ∧ ∧ (возврат забракованной части z_i на устранение брака начальнику монтажного участка 4А на время [$Ti * b_{i/} Ni$] + 1) (0)	Считаем, что весь заказ целиком ждет исправления брака, это потеря времени
7	($b_i = 0$, т.е. продукция z_i прошла контроль качества)	(сдача z_i на склад готовой продукции) (0) ∧ (упаковка z_i) (1) ∧ (предпродажная подготовка z_i) (0) ∧ (доставка заказа потребителю 1А)(1)	

Окончание табл. 4.8

1	2	3	4
8	(от начальника монтажного участка 4А получено сообщение, что устранен брак b_i, т. е. продукция z_i прошла контроль качества)	(сдача z_i на склад готовой продукции) (0) $\wedge$ (упаковка z_i) (1) $\wedge$ (предпродажная подготовка z_i) (0) $\wedge$ (доставка заказа потребителю 1А)(1)	
9	(заказ потребителю 1А доставлен) $\wedge$ $\wedge$ ($T_i^* \le T_i$)	(менеджеру 2А послано сообщение, что заказ *Zi* выполнен)$\wedge$($T_i^* \le T_i$)	
10	(заказ потребителю 1А доставлен) $\wedge$($T_i^* > T_i$)	(менеджеру 2А послано сообщение, что заказ *Zi* выполнен)$\wedge$($T_i^* > T_i$)	
11	От потребителя 1А поступила претензия в течение гарантийного срока	(исправление брака)(0)	
12	От потребителя 1А поступила претензия после гарантийного срока	(исправление брака)(0) $\wedge$ (сообщение менеджеру 2А о информации для оформления доп. соглашения к договору)(0)	

Таблица 4.9

Агент 4А: «Начальник монтажного участка»

ЦельАгента 4А: Выполнение заказов точно в срок, повышение производительности труда

№ п/п	Если	То	Описание
1	от начальника производственного участка 3А получено сообщение о начале выполнения заказа *Zi*	(выполнение заказа *Zi*)(*Ti*) ∧ (начальнику производственного участка 4А сообщение о выполнении заказе *Zi, Ni, Xi, Pi, Ti*)(*Ti*)	Считаем, что заказу *Zi* сопутствует вся информация: *Ni, Xi, Ti, Si, Pi, Ki* и т. д. Считаем, что прямые затраты (материалы, зарплата рабочих, коммунальные платежи участка, транспорт участка и т. д.) учтены в прямой себестоимости продукции Считаем, что на складе материалов есть некоторый запас, позволяющий начинать производство не дожидаясь поступления материалов и комплектующих
2	(от начальника производственного участка 3А возврат забракованной части z_i)	(устранение брака на время $[Ti * b_{i/} Ni] + 1)$ $(Ti * b_i / Ni)$ ∧ (начальнику производственного участка 4А сообщение об устранении брака *Zi, Ni, Xi, Pi, Ti*)($Ti * b_i / Ni$)	
3	от начальника производственного участка 3А возврат по гарантийному или послегарантийному обслуживанию части z_i)	(устранение брака)(2)∧ (начальнику производственного участка 4А сообщение об устранении брака)(2)	

Таблица 4.10

Агент 5А: Класс объектов «Конкурент». Индивидуальные характеристики берутся из матрицы *Eij*
Цель: Вытеснение предприятия УИГ с рынка

№ п/п	Если	То	Описание
1	От начальника отдела продаж 2А поступило сообщение, что предприятие изменяет цену	m- номер экземпляра класса конкурента, от 1 до 9 (каждый конкурент – *m* вызывает процедуру Proc1 определения цены) (E*m* 5) ∧ (Сообщение начальнику отдела продаж 2А, что Конкурент m изменяет цену)(E*m* 5)	Предполагается, что модель конкурента – класс объектов – агентов – игроков рынка за исключением нашего предприятия. Параметры конкурента определяются из таблицы *Eijm* от 1 до 9 После изменения цены начальником отдела продаж 2А, каждый конкурент от 1 до 9 вызывает процедуру Proc1 определения цены для корректировки своей цены через время реакции, описанное в матрице *Em* 5 (*m*-я строка, 5-й столбец)

Таблица 4.11

Процедура определения цены при стратегии достижения большей прибыли путем увеличения индивидуальной доли рынка
Proc1 – запускается каждым из конкурентов, входные параметры:

№ п/п	Обозначение	Наименование параметра	Поведение параметра
1	m	Номер конкурента	
2	Em1	Доля рынка *m*-го конкурента	Меняется во время моделирования
3	Цmin	Предельная минимальная цена, ниже компания разоряется	
4	eps	Окрестность Цmin (предельной минимальной цены), в которую опасно входить	
5	Em7	Действующая цена m-го конкурента	Меняется во время моделирования
6	kol	Количество участников рынка	
7	Em8	Персональный признак стратегии фирмы: 1-активная, рискованная; 2-пассивная, осторожная	

Таблица 4.12

База знаний

№ п/п	Если	То	Описание
1	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ ∧ (*Em*7 − Цmin < eps) ∧ (*Em*8 = 1)	*Em*7 = 97 % * * max(*E*17, *E*27,..., *E*kol 7)	Действующая цена находится в окрестности Цmin. Конкурент активный. Увеличение цены до уровня на 3% ниже самой высокой цены предшествующего периода
2	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ ∧ (*Em*7 − Цmin < eps) ∧ (*Em*8 = 2)	*Em*7 = 101 % * **Em*7	Действующая цена находится в окрестности Цmin. Конкурент пассивный. Увеличение цены на 1 % по сравнению с предыдущим периодом
3	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ (*Em*7 − Цmin > = eps) ∧ ∧ ((160 % / kol) < *Em*1 < = 100 %)	E*m*7 = E*m*7	Действующая цена находится вне окрестности Цmin. Доля рынка – очень высокий уровень. Сохранение прежней цены без изменений
4	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ ∧ (*Em*7 − Цmin > = eps) ∧ ((100 % /kol) < *Em*1 < = < = ((160 % /*kol*)) ∧ (*Em*8 = 1)	E*m*7 = 97 % * * max (E17, *E*27,..., *Ekol* 7)	Действующая цена находится вне окрестности Цmin. Доля рынка – высокий уровень. Конкурент активный. Увеличение цены до уровня на 3% ниже самой высокой цены предшествующего периода
5	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ (*Em*7 − Цmin > = eps) ∧ ∧ ((100 % / *kol*) < *Em*1 < = ((160 % / *kol*)) ∧ ∧ ∧ (*Em*8 = 2)	*Em*7 = = 101 % **Em*7	Действующая цена находится вне окрестности Цmin. Доля рынка – высокий уровень. Конкурент пассивный. Увеличение цены на 1% по сравнению с предыдущим периодом.
6	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ (*Em*7 − Цmin > = eps) ∧ ∧ ((40 % / *kol*) < = *Em*1 < = ((100 % / *kol*))	*Em*7 = = 99 % * *Em*7	Действующая цена находится вне окрестности Цmin. Доля рынка – средний или низкий уровень. Уменьшение цены на 1 % по сравнению с предыдущим периодом.
7	(Вызвана процедура Proc1 определения цены *m*-м конкурентом) ∧ (*Em*7 − Цmin > = eps) ∧ ∧ (0 % < = *Em*1 < = (40 % / *kol*))	*Em*7 = 103 % * * min(*E*17, *E*27,..., *Ekol* 7)	Действующая цена находится вне окрестности Цmin. Доля рынка – очень низкий уровень. Уменьшение цены до уровня на 3 % выше самой низкой цены предшествующего периода.

Был проведен ряд экспериментов, направленных на поиск эффективной ценовой стратегии и учитывающих различные сочетания поведений агентов-конкурентов (активность / пассивность). На рис. 4.9 представлены графики экспериментов ценовых стратегий (рис. 4.10), приводящих к вытеснению двух небольших предприятий с рынка.

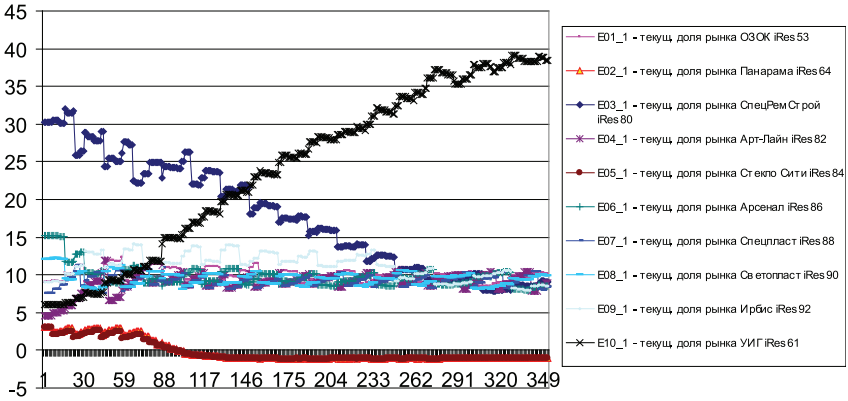


Рис. 4.9. Вытеснение двух предприятий с рынка при пассивном поведении конкурентов

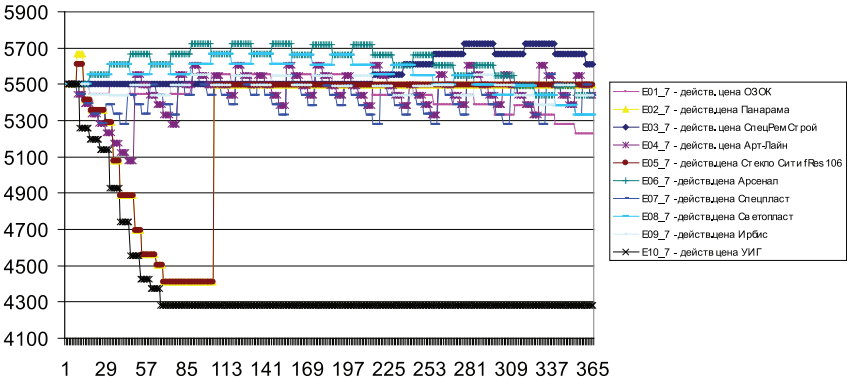


Рис. 4.10. Ценовые стратегии, приводящие к вытеснению предприятий

На рис. 4.11–4.12 представлены графики зависимостей доли рынка и цены при активных моделях поведения конкурентов и скачкообразной ценовой стратегии ЗАО «УИГ», в результате которой

предприятие увеличивает свой сегмент за счет лидирующей роли в диктовании цены и за счет того, что конкуренты с большим сегментом имеют большее время реакции по цене и не успевают оперативно реагировать на ее изменение.

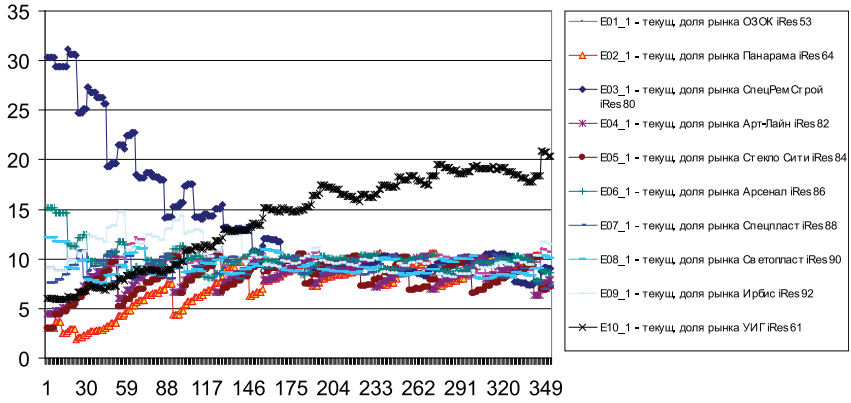


Рис. 4.11. Изменение долей рынка при скачкообразной ценовой стратегии и активном поведении конкурентов на рынке

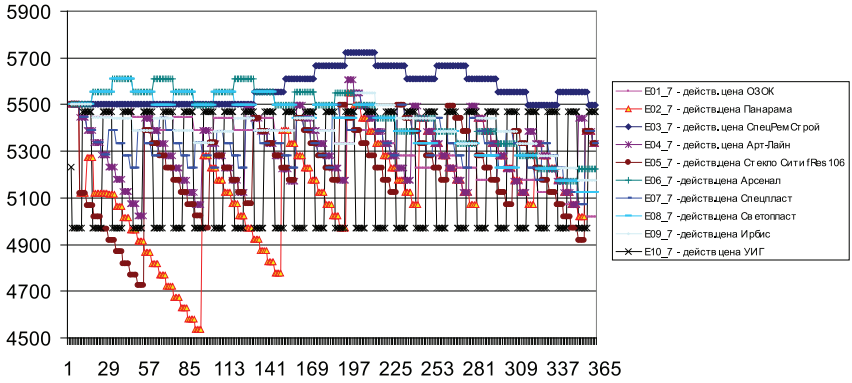


Рис. 4.12. Реакции конкурентов на скачкообразное изменение цены

При активном поведении конкурентов и нескачкообразном изменении цены существенный прирост сегмента рынка не наблюдается (рис. 4.13–4.14).

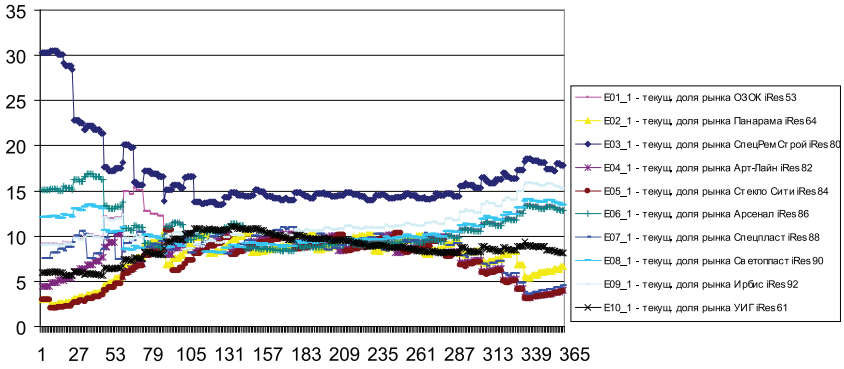


Рис. 4.13. Перераспределение долей конкурентов при активных моделях поведения и нескачкообразном изменении цены

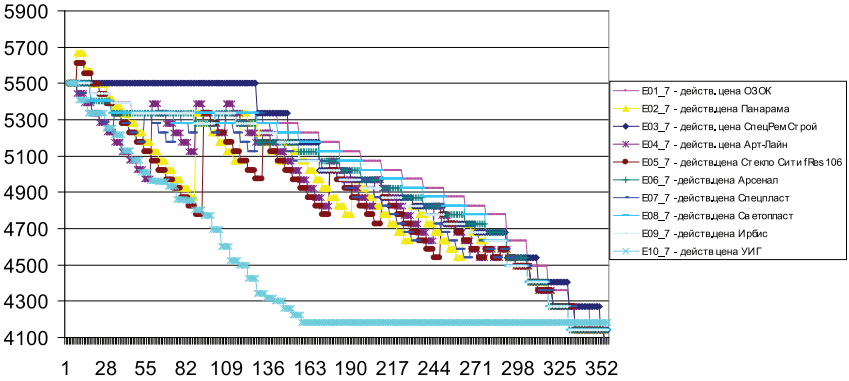


Рис. 4.14. Реакции активных конкурентов на снижение цены

На рис. 4.15 представлены результаты изменения прибыли ЗАО «УИГ» в зависимости от различных ценовых стратегий. Графики «Ряд6» и «Ряд7» – результаты экспериментов, связанные с применением скачкообразного изменения цены и активных моделей поведения конкурентов. Графики изменения цены и доли рынка

для «Ряд7» представлены ранее на рис. 4.11 и 4.12. Графики «Ряд2» и «Ряд5» показывают также неплохие показатели динамики прибыли, но относятся к экспериментам, опирающимся на пассивные модели поведения конкурентов.

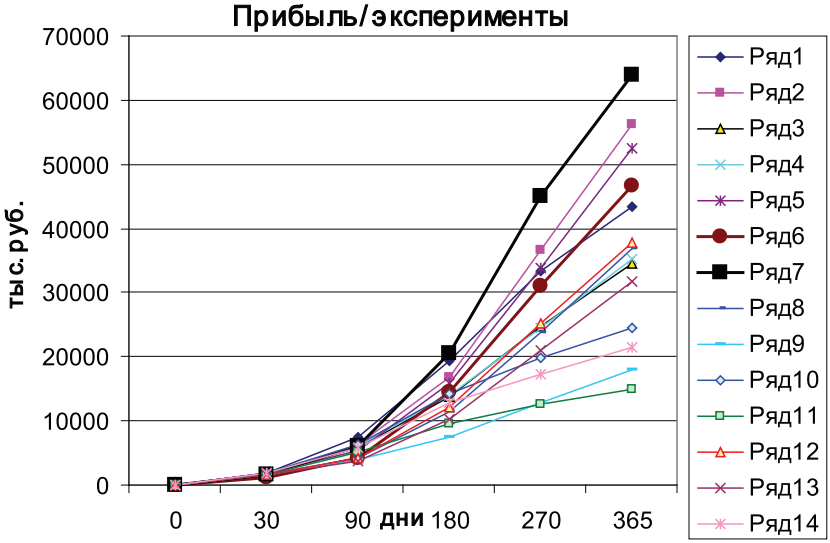


Рис. 4.15. Показатели прибыли при различных ценовых стратегиях

На рис. 4.16 представлены результаты изменения доли рынка ЗАО «УИГ» в зависимости от различных ценовых стратегий. Эксперименты «Ряд6» и «Ряд7» с применением скачкообразного изменения цены и активных моделей поведения конкурентов соответствуют показателям годовой прибыли 46 и 63 млн руб. и повышению доли рынка соответственно до 22 и 20 %. Эксперименты «Ряд2», «Ряд5» и«Ряд13» показывают хорошие результаты по доле рынка, но учитывают пассивные модели поведения конкурентов, на что не приходится рассчитывать. Эксперимент «Ряд8» с активными моделями поведения конкурентов не удовлетворяет по показателям прибыли.

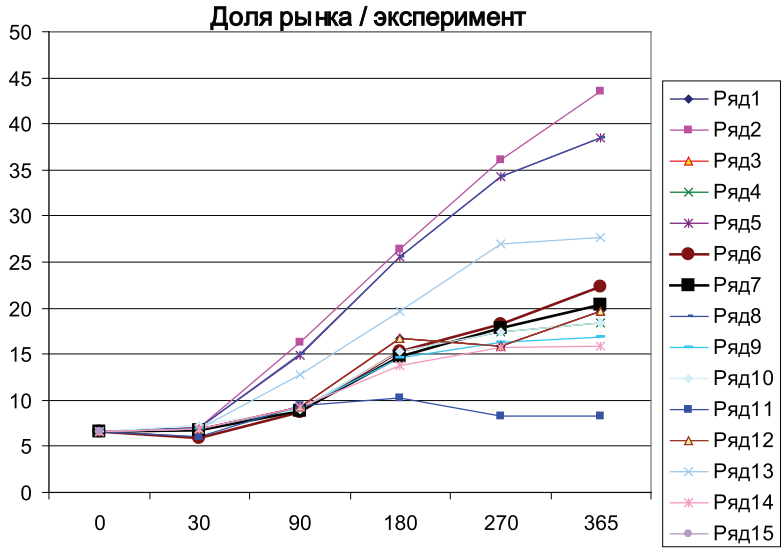


Рис. 4.16. Показатели долей рынка при различных ценовых стратегиях

На рис. 4.17–4.18 представлены графики зависимости количества монтажных звеньев и количества центров продаж с учетом выбранной ценовой стратегии.

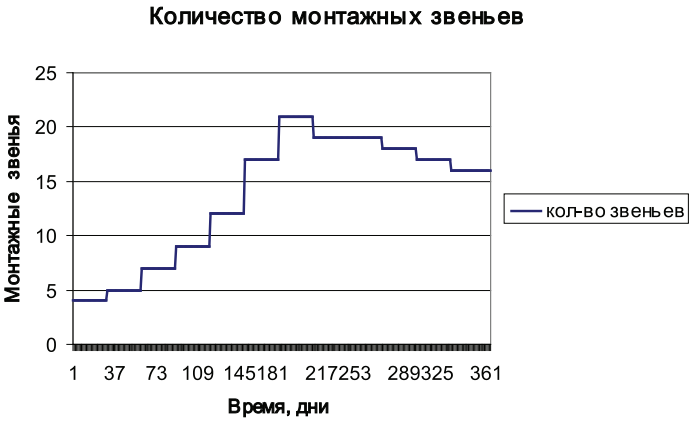


Рис. 4.17. Зависимость количества монтажных звеньев от времени с учетом объема продаж

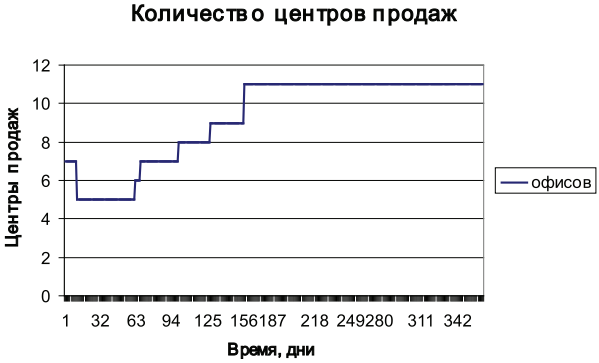


Рис. 4.18. Зависимость количества центров продаж от времени с учетом объема продаж

Таким образом, были проведены серии экспериментов, позволившие определить ценовую стратегию, следуя которой можно увеличить в течение года долю рынка с 6,6 до 20–22 %. В рамках данной задачи также были найдены оптимальные значения количеств монтажных звеньев и центров продаж в зависимости от объема продаж в месяц. Прогнозируемый экономический эффект составляет 46 млн. руб. в год.

Выводы

1. Как следует из данной главы, на основе результатов проведенных экспериментов и результатов внедрения теоретические результаты прошли проверку в условиях действующих предприятий, что подтвердило правильность и обоснованность разработанных положений и выводов.

2. СДМС BPsim2 работоспособна и обеспечивает возможность:

– создания сложных ситуационных моделей мультиагентных процессов преобразования ресурсов;

– проведения структурного и параметрического синтеза модели мультиагентного процесса преобразования ресурсов;

– формирования отчетов по структуре процесса;

– формирования отчетов по результатам экспериментов;

– экспорта результатов экспериментов во внешние средства анализа данных;

– оценки состояния ресурсов и средств, временных и стоимостных характеристик процесса преобразования ресурсов, а также производных и консолидированных параметров;

– проведения функционально-стоимостного анализа процесса преобразования ресурсов.

ЗАКЛЮЧЕНИЕ

Таким образом, в рамках данной работы получены следующие результаты:

1. Определен перечень характеристик и проведен сравнительный анализ наиболее распространенных проблемно-ориентированных систем, близких к СДМС: AnyLogic, ARIS, G2. К недостаткам названных систем можно отнести: неполный набор функциональных возможностей мультиагентной СДМС; отсутствие поддержки функций проектирования концептуальной модели предметной области и построения мультиагентных моделей, содержащих интеллектуальных агентов; недостаточную проблемную ориентацию на мультиагентные процессы преобразования ресурсов; ограниченную поддержку русского языка; ориентированность на программирующего пользователя.

2. Разработаны требования к ситуационной математической модели мультиагентного процесса преобразования ресурсов, которая должна обеспечивать следующие функции: моделирование динамических процессов преобразования ресурсов; наличие сообществ интеллектуальных агентов, участвующих в управлении процессом преобразования ресурсов; применение ситуационного подхода.

3. За основу математической модели взята модель дискретного процесса преобразования ресурсов. В рамках разработанной математической модели:

- определены основные объекты моделирования мультиагентных процессов преобразования и их характеристики (операции, ресурсы, средства, заявки, очереди заявок, сигналы, сообщения, процессы, источники и приемники ресурсов, перекрестки, параметры, цели, агенты), а также система выполняемых ими действий;
- рассмотрены типовые математические модели (семиотическая модель Клыкова Ю.И. и SIE-модель Филипповича А.Ю.) для построения математической ситуационной модели мультиагентного процесса преобразования ресурсов;
- за основу построения модели интеллектуального агента принята ОМИА (Швецов А.Н.), которая модернизирована и доработана применительно к проблемной области процессов преобразования ресурсов;

● в качестве модели представления знаний выбрано фреймово-семантическое представление на основе фрейм-концептов и концептуальных графов, позволяющее получить концептуальную модель предметной области;

● решена задача перехода (совмещения) модели представления знаний, концептуальной модели и их технической реализации на уровне реляционной базы данных; показано, что данный подход позволяет использовать язык Transact-SQL при построении модели предметной области, вводе данных и знаний, реализации механизма логического вывода;

● разработаны механизм вывода мультиагентного процесса преобразования ресурсов, типы правил, алгоритм работы интеллектуального агента и алгоритм ситуационно-имитационного моделирования.

4. На основе математической модели разработаны:

● интерфейсы СДМС, ориентированные на конечного пользователя;

● программное, информационное, алгоритмическое и методическое обеспечение проблемно-ориентированного пакета BPsim2;

● технология работы с ППП BPsim2.

5. Разработана СДМС BPsim2, обладающая полным перечнем функциональных возможностей, предъявляемых к проблемно-ориентированному ППП динамического моделирования мультиагентных процессов преобразования ресурсов, и отличающаяся:

● полным набором функциональных возможностей мультиагентной СДМС;

● поддержкой функции проектирования концептуальной модели предметной области;

● возможностью построения мультиагентных моделей, содержащих ИА;

● проблемной ориентацией на мультиагентные процессы преобразования ресурсов;

● интеграцией с методикой BSC;

поддержкой русского языка.

6. Созданная СДМС «BPsim2» внедрена в департаменте оконных конструкций предприятия ЗАО «Уральская индустриальная группа». Внедрение в ЗАО «УИГ» позволило определить ценовую стратегию, следуя которой можно увеличить в течение года долю рынка с 6,6 до 20–22 % (экономический эффект составляет 46 млн руб. в год).

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

BSC	Balanced Score Card – система сбалансированных показателей
CASE	Computer Aided Software Engineering – автоматизированное проектирование программного обеспечения
EPC	Extended Event Driven Process Chain – расширенная нотация описания цепочки процесса, управляемого событиями
UML	Unified Modeling Language – унифицированный язык моделирования
АП	Абсолютный приоритет
АСС	Аналитическая ситуационная система
АСУ	Автоматизированная система управления
АСЦ	Аналитический ситуационный центр
БВСЦ	Блок выбора сценария
БД	База данных
БЗ	База знаний
ДСС	Дискретные ситуационные сети
ЗА	Значение атрибута
ИА	Интеллектуальный агент
ИК	Имя концепта
ИМ	Имитационное моделирование (имитационная модель)
ИМВИА	Имитационная модель взаимодействия интеллектуальных агентов
ИНС	Интеллектуальная система
ИО	Информационный объект
ИП	Информация о применении
ИС	Информационная система
ИТ	Информационные технологии
ИФ	Имя фрейма
КГ	Концептуальный граф
КМПО	Концептуальная модель предметной области
КО	Концептуальное отношение
КУ	Коэффициент уверенности

ЛПР	Лицо, принимающее решения
МО	Множество определения
МТР	Материально-технические ресурсы
ООП	Объектно-ориентированный подход
ОТС	Организационно-техническая система
ПК	Персональный компьютер
ПО	Программное обеспечение
ППП	Пакет прикладных программ
ППР	Процесс принятия решений
РБД	Реляционная база данных
РП	Рабочая память
РССОИ	Распределенные ССОИ
СА	Структура атрибутов
СК	Структура концептов
СИМ	Система имитационного моделирования
СДМС	Система динамического моделирования ситуаций
СИО	Структурный информационный объект
СМ	Ситуационная модель
СППР	Система поддержки принятия решений
ССЛ	Структура слотов
ССМ	Система ситуационного моделирования
ССОИ	Система ситуационного отображения информации
ССП	Структура сценариев поведения
ССУ	Система ситуационного управления
СУБД	Система управления базами данных
СЦ	Ситуационный центр
СЦО	Ситуационный центр наблюдения (отображения)
СЭМ	Система экспертного моделирования
ТРА	Теория речевых актов
ТФ	Тип фрейма
ТЭП	Технико-экономического проектирование
ФК	Фрейм-концепт
ФЭ	Фрейм-экземпляр
ХД	Хранилище данных
ЭС	Экспертная система
ЯВУ	Язык высокого уровня
ЯПЗ	Язык представления знаний

СПИСОК ЛИТЕРАТУРЫ

1. Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика / А.В. Борщев // ExponentaPro. 2004.№ 3–4.

2. Имитационное моделирование производственных систем / под общ. ред. А.А. Вавилова. М.: Машиностроение; Берлин: Техник, 1983. 416 с.

3. Технология системного моделирования / Е.Ф. Аврамчук [и др.]; под общ. ред. С.В. Емельянова. М.: Машиностроение; Берлин: Техник, 1988. 520 с.

4. Интеллектуальные информационные технологии в управленческой деятельности // III Международный научно-практический семинар: сборник материалов / под ред. С.Л. Гольдштейна. Екатеринбург: ИПК УГТУ-УПИ, 2001. 368 с.

5. Системная интеграция в управленческой деятельности: сборник статей / под ред. С.Л. Гольдштейна. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2002. 309 с.

6. Представление и использование знаний: пер. с япон./под ред. Х. Уэно, М. Исидзука. М.: Мир, 1989. 220 с.

7. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю.Г. Карпов. СПб.: БХВ-Петербург, 2005. 400 с.: ил.

8. Клыков Ю.И. Ситуационное управление большими системами / Ю.И. Клыков. М.: Энергия, 1974. 136 с.

9. Клыков Ю.И. Семиотические основы ситуационного управления / Ю.И. Клыков. М.: МИФИ, 1974. 220 с.

10. Клыков Ю.И. Банки данных для принятия решений / Ю.И. Клыков, **Л.Н.** Горьков. М.: Сов. радио, 1980. 155 с.

11. Статические и динамические экспертные системы: учеб. пособие / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. М.: Финансы и статистика, 1996. 320 с.

12. Поспелов Д.А. Мышление и автоматы / Д.А. Поспелов, В.Н. Пушкин. М.: Советское радио, 1972.224 с.

13. Поспелов Д.А. Ситуационное управление: теория и практика / Д.А. Поспелов. М.: Наука, 1986. 288 с.

14. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II: пер. с англ. / А. Прицкер.М.: Мир, 1987. 646 с.

15. Советов Б.Я. Моделирование систем: учеб. для вузов. 3-е изд. / Б.Я. Советов, С.А. Яковлев. М.: Высш. шк., 2001. 343 с.

16. Форрестер Дж. Основы кибернетики предприятия (Индустриальная динамика): пер. с англ. / Дж. Форрестер; под ред. Д.М. Гвишиани. М.: Прогресс, 1971. 340с.

17. Форрестер Дж. Мировая динамика: пер. с англ. /Дж. Форрестер; под ред. Д.М. Гвишиани, Н.Н. Моисеева. М.: Наука, 1978. 168 с.

18. Филиппович А.Ю. Интеграция ситуационного, имитационного и экспертного моделирования в полиграфии / А.Ю. Филиппович. М., 2003. 310 с.

19. Филиппович А.Ю. Система имитационного моделирования допечатных процессов / А.Ю. Филиппович, И. Шапиро // Проблемы полиграфии и издательского дела. 2002. № 3. С. 62-75.

20. Филиппович А.Ю. Обучающие ситуационные центры / А.Ю. Филиппович // Системный администратор. 2003. № 4.

21. Швецов А.Н. Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений: дис. ... д-ра техн. наук: 05.13.01 / А.Н. Швецов. Санкт-Петербург, 2004. 461 с.

22. Дианов С.В. Архитектура мультиагентной системы организационного управления / С.В. Дианов, А.Н. Швецов // Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и систем искусственного интеллекта: матер. науч.-техн. конф. Вологда: ВоГТУ, 2001. С. 104–108.

23. Яковлев С.А. Архитектура баз знаний в распределенных интеллектуальных информационных системах / С.А. Яковлев, А.Н. Швецов // Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и систем искусственного интеллекта: матер. науч.-техн. конф. Вологда: ВоГТУ, 2001. С. 124–128.

24. Шеер А.В. Бизнес-процессы. Основные понятия. Теория. Методы / А.В. Шеер. М.: Весть Метатехнология, 1999. 182 с.

25. Шеер А.В. Моделирование бизнес-процессов / А.В. Шеер. М.: Весть-Метатехнология, 2000. 205с.

26. Wooldridge M. IntelligentAgent: Theory and Practice / M. Wooldridge, N. Jennings // Knowledge Engineering Review. 1995. № 10 (2).

27. Greenwald A. Guest Editors' Introduction: Agents and Markets / A. Greenwald, N. Jennings, P. Stone // Intelligent Systems. 2003. Vol. 18. p. 12–14.

28. Dash R. Computationals-Mechanism Design: A Call to Arms / R. Dash, D. Parkes, N. Jennings // Intelligent Systems. 2003. Vol. 18. p. 40-47.

29. Minsky M. A framework for Representing Knowledge in The Psychology of Computer Vision, P.H. Winston (ed.), McGraw-Hill 1975.

30. Model Checking Rational Agents /R. Bordini, M. Fisher, W. Visser, M. Wooldridge // Intelligent Systems. – 2003. Vol. 18. p. 40-47.

31. Злобин Э.В. Управление качеством в образовательной организации / Э.В. Злобин, С.В. Мищенко, Б.И. Герасимов. Тамбов: ИздвоТамб. гос. техн. ун-та, 2004. 88 с.

32. Падучева Е.В. Семантические типы ситуаций и значение всегда / Е.В. Падучева // Семантика и информатика. 1985. Вып. 24. С. 96–116.

33. Аксенов К.А. Исследование и разработка средств имитационного моделирования дискретных процессов преобразования ресурсов: дис. ... канд. техн. наук: 05.13.18 / К.А. Аксенов. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2003. 188 с.

34. Аксенов К.А. Принципы построения системы имитационного моделирования процессов преобразования ресурсов BPsim / К.А. Аксенов, Б.И. Клебанов // Материалы первой Всероссийской научн.-практ. конф. «Опыт практического применения языков и программных систем имитационного моделирования в промышленности и прикладных разработках»: сборник докладов. Санкт-Петербург: ФГУП ЦНИИ технологии судостроения, 2003. Т. 1. С. 36-40.

35. Aksyonov K. Computer-aided design system of simulation business process model / K.A. Aksyonov, B.I. Klebanov, A.A. Hrenov // Proceedings of the 4th IMACS Symposium on Mathematical Modeling, ARGESIM Report № 24. Austria, VieenaUniversity of Technology. 2003. P.1414–1420.

36. Аксенов К.А. Система имитационного моделирования процессов преобразования ресурсов / К.А. Аксенов, Б.И. Клебанов // Научные труды IV отчетной конф. молодых ученых ГОУ ВПО УГТУ-УПИ:сборник статей. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2003. Ч. 1. С. 135-136.

37. Аксенов К.А. Разработка и применение средств имитационного моделирования процессов преобразования ресурсов BPsim / К.А. Аксенов, Б.И. Клебанов // 7-я Московская международная телекоммуникационная конференция студентов и молодых ученых. М.: МИФИ, 2003.Режим доступа: http://molod.mephi.ru/Data/358.htm – электронная публикация.

38. Аксенов К.А. Исследование типовых математических схем для моделирования процессов преобразования ресурсов / К.А. Аксенов, Б.И. Клебанов // 7-я Московская международная телекоммуникационная конференция студентов и молодых ученых. М.: МИФИ, 2003.Режим доступа: http://molod.mephi.ru/Data/384.htm-электронная публикация.

39. Аксенов К.А. Применение системы BPsim для анализа деятельности строительного предприятия / К.А. Аксенов, Е.Ф. Смолий, С.Ю. Долматов // Информационные технологии и электроника: материалы VIII Всероссийской студенческой науч.-техн. конф. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2003. Режим доступа: http://webconf.nexcom.ru – электронная публикация.

40. Аксенов К.А. Исследование адекватности расширенной сети Петри процессу преобразования ресурсов / К.А. Аксенов // Восьмая Всероссийская студенческая научно-техническая Интернет-конференция «Инфор-мационные технологии и электроника». 2003. Режим доступа: http://webconf.nexcom.ru – электронная публикация.

41. Аксенов К.А. Проблемно-ориентированная система имитационного моделирования процессов преобразования ресурсов / К.А. Аксенов, Б.И. Клебанов, Е.Ф. Смолий // Научные труды международной научно-практической конференции «СВЯЗЬ-ПРОМ 2004» в рамках 1-го Евро-Азиатского международного форума «СВЯЗЬ-ПРОМЭКСПО 2004». Екатеринбург: ЗАО «Компания Реал-Медиа», 2004. С. 40-51.

42. Аксенов К.А. Создание системы имитационного моделирования процессов преобразования ресурсов Bpsim / К.А. Аксенов, Б.И. Клебанов // Проектирование и анализ радиотехнических и информационных систем: Серия радиотехническая // Вестник УГТУ-УПИ. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2004. № 18 (48). С. 174–182.

43. Аксенов К.А. Проблемно-ориентированная система имитационного моделирования процессов преобразования ресурсов: Информационные системы в технике и образовании: Серия радиотехническая / К.А. Аксенов, Б.И. Клебанов, Е.Ф. Смолий // Вестник УГТУ-УПИ. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2004. № 19 (49). С. 20-32.

44. Аксенов К.А. Применение средств имитационного моделирования в системе стратегического управления вуза / К.А. Аксенов, Б.И. Клебанов, Н.В. Гончарова // 2-я международная конференция

«Стратегическое управление и институциональные исследования в высшем образовании». Режим доступа: www.studiorum.ru – 2003 электронная публикация.

45. Аксенов К.А. Имитационное моделирование в стратегическом управлении вуза / К.А. Аксенов, Б.И. Клебанов, Н.В. Гончарова // Вестник науки Костанайской социальной академии. Костанай, 2003. № 5. С. 24–28.

46. Аксенов К.А. Применение средств имитационного моделирования в системе стратегического управления вузом / К.А. Аксенов, Б.И. Клебанов, Н.В. Гончарова // Журнал «Университетское управление: практика и анализ». Екатеринбург, 2004. № 2 (30). С. 54–57.

47. Статические и динамические модели процессов вуза / К.А. Аксенов, М.Н. Клепинина, Б.И. Клебанов, Н.В. Гончарова, А.И. Боков, Д.А. Якусов // VIII Всероссийская студенческая науч.-техн. конф. «Информационные технологии и электроника». Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2003. Режим доступа: http://webconf.nexcom.ru/file.php/3/moddata/forum/3/12/_2.doc – электронная публикация.

48. Кононенко И.А. Базовые программно-языковые средства моделирования дискретно-линейных процессов: дис. ... канд. техн. наук: 05.13.12 / И.А. Кононенко / Урал. науч. центр. инс-т математики и механики. Свердловск, 1981. 210 с.

49. Ситников И.О. Средства иерархического моделирования в системе автоматизированного проектирования дискретных устройств: дис. ... канд. техн. наук: 05.13.12 / И.О. Ситников / Урал. науч. центр. инс-т математики и механики. Свердловск, 1983. 190 с.

50. Захарова Г.Б. Интеллектуализация средств логико-компилятивного проектирования РЭА: дис. ... канд. техн. наук: 05.13.12 / Г.Б. Захарова / Рос. акад. наук. Урал. отд-ние. Инс-т машиноведения. Екатеринбург, 1993. 171 с.

51. Старцев М.А. Интегрированная информационная система для автоматизированного управления процессом капитального строительства на промышленном предприятии на основе иерархических ситуационных моделей сетевого планирования: автореф. дис. ... канд. техн. наук: 05.13.06 / М.А. Старцев / Уфим. гос. авиац. техн. ун-т. Уфа, 1999. 16 с.

52. Трахтенгерц Э.А. Компьютерные системы поддержки принятия управленческих решений / Э.А. Трахтенгерц // Сборник статей II международной конференции по проблемам управления. М., 2003.

53. Трахтенгерц Э.А. Компьютерная поддержка принятия решений / Э.А. Трахтенгерц. М.: СИНТЕГ, 1998.

54. Мильнер Б.З. Теория организаций / Б.З. Мильнер. М.: ИНФРА-М, 1998. 336 с.

55. Савина О.А. Управление промышленными предприятиями с использованием адаптивных систем имитационного моделирования: автореф. дис. ... д-ра экон. наук: 08. 00. 05, 08. 00. 13 / О.А. Савина / Орлов. гос. техн. ун-т. Орел, 2001. 45 с.

56. Ginzberg M.J. Decision support systems: Issues and perspectives. – In: Decision support systems / Ed. by M.J. Ginzberg, W.R. Itman, E.A. Stohr. Amsterdam: North-Holland, 1982, p. 9-31.

57. Борисов А.Н. Диалоговые системы принятия решений на базе мини-ЭВМ: Информационное, математическое и программное обеспечение / А.Н. Борисов, Э.Р. Вилюмс, Л.Я. Сукур. Рига: Зинатне, 1986. 195 с.

58. Ларичев О.И. Качественные методы принятия решений / О.И. Ларичев, Е.М. Мошкович. М.: Наука; Физматлит, 1996.

59. Simonovic A. Decision support for sustainable water resources development in water resources planning in a changing world / A. Simonovic, P. Slobodan // Proceeding of International UNESCO symposium, Karlsruhe, Germany, part III. p. 3-13,1994.

60. Ginzberg M.J. A decision support: Issues and Perspectives / M.J. Ginzberg, E.A. Stohr / Processes and Tools for Decision Support. Amsterdam, North – Holland Publ. Co, 1983.

61. Ольве Н.-Г. Оценка эффективности деятельности компании: Практическое руководство по использованию сбалансированной системы показателей / Н.-Г. Ольве, Ж. Рой, М. Ветер. М.: Вильямс, 2004. 304 с.

62. Ерматель – Система учета и планирования ресурсов предприятия. Презентация системы. – 2002.

63. TS-Полиграфия – система автоматизации. Краткое описание системы. [Электронный ресурс]. Режим доступа: http://vip.b2bsbn.ru.

64. Краткое описание системы. [Электронный ресурс]. Режим доступа: http://www.pole.tl.ru/add/sitcenter-full.htm.

65. Краткое описание системы. [Электронный ресурс]. Режим доступа: http://iu5.bmstu.ru/~philippovicha/Articles/Sit_Centres.htm.

66. Организация взаимодействия человека с техническими средствами АСУ. В 7 кн. Кн. 4. Отображение информации: практическое пособие / В.М. Гасов, А.И. Коротаев, С.И. Сенькин; под ред. В.Н. Четверикова. М.: Высш. шк., 1990. 111 с.

67. IBS Центр принятия решений – СЦ Министерства природных ресурсов РФ. Краткое описание системы. [Электронный ресурс]. Режим доступа: www.tek.ibs.ru

68. Мобильный пункт управления для МЧС. Описание системы. [Электронный ресурс]. Режим доступа: www.proekt-technica.ru

69. Ситуационно-кризисный центр Минатома РФ. Описание системы. [Электронный ресурс]. Режим доступа: www.minatom.ru

70. БалтикСофт. Ситуационный центр «КАНТ». Описание системы. [Электронный ресурс]. Режим доступа: http://www.baltsoft.ru/podrob1.htm

71. Триумф-аналитика. Описание системы. [Электронный ресурс]. Режим доступа: www.ta.interrussoft.com, 2002

72. Чернов А.Г. Ситуационный центр как инструмент регулирования энергетики:Краткое описание системы / А.Г. Чернов. [Электронный ресурс]. Режим доступа: http://esco-ecosys.narod.ru/2004_1/reports/29.doc

73. Кельтон В. Имитационное моделирование. Классика CS. 3-е изд. / В. Кельтон, А. Лоу. СПб.: Питер; Киев: Издательская группа BHV, 2004. 847 с.

74. Гнеденко Б.Д. Введение в теорию массового обслуживания / Б.Д. Гнеденко, И.Н. Коваленко. М.: Наука, 1987. 336 с.

75. Edward J. Williams, Ali Gunal. Supply chain simulation and analysis with SIMFLEX™ // 2003 Winter Simulation Conference (WSC'03), December 8–11, 2002, San Diego, California, USA pp. 231–237.

76. Сергеева И.Г. Интеллектуальное управление производственно-экономической системой в условиях неопределенности на основе имитационного моделирования: автореф. дис. ... канд. техн. наук: 05.13.10 / И.Г. Сергеева / Уфим. гос. авиац. техн. ун-т. Уфа, 2000. 16 с.

77. AnyLogic. Учебное пособие по агентному моделированию. Режим доступа: www.xjtek.com

78. Arena улучшает возможности для бизнеса в условиях новой экономики. Режим доступа: http://www.interface.ru/sysmod/sysmodh.htm – электронная публикация.

79. Маклаков С. Имитационное моделирование с Arena / С. Маклаков. Компьютер-пресс. 2001. № 7. С. 135-136.

80. ARIS – основы теории Режим доступа: http://icc.migsv.ru/ibm866/citforum/- электронная публикация.

81. Курс ReThink. Конспект лекций. М.: АО «АРГУССОФТ КОМПАНИЯ», 1996. 87 с.

82. Описание системы G2. Режим доступа: www.gensym.com.

83. Яковлев С.А. Имитационное моделирование поведения интеллектуального агента в корпоративных системах поддержки принятия решений / С.А. Яковлев, А.Н. Швецов // Конференция ИММОД 2003. [Электронный ресурс]. Электрон. дан. 2003. Режим доступа: www.gpss.ru.

84. Эрлих А.И. Проблемы моделирования в интеллектуальных системах / А.И. Эрлих // Вестник МГТУ: Сер. «Приборостроение». 1994. № 1. С. 29-33.

85. Павловский Ю.Н. Имитационные модели и системы / Ю.Н. Павловский. М.: ФАЗИС: ВЦ РАН, 2000. 134 с.

86. Гаврилова Т.А. Состояние и перспективы разработки баз знаний интеллектуальных систем / Т.А. Гаврилова // Журнал «Новости искусственного интеллекта». М., 1996. № 1. С.5-43.

87. Косов С.В. Разработка экспертной системы для управления давлением в системах водоснабжения / С.В. Косов, В.В. Девятков // Вестник МГТУ. Сер. «Приборостроение». 1996. № 2. С. 34–45.

88. Тарасов В.Б. Развитие прикладных интеллектуальных систем: анализ основных этапов, концепций и проблем / В.Б. Тарасов, Н.М. Соломатин // Вестник МГТУ. Сер. «Приборостроение». 1994. № 1. С. 5-15.

89. Кузнецов И.П. Кибернетические диалоговые системы / И.П. Кузнецов. М.: Наука, 1976. 293 с.

90. Искусственный интеллект. Кн. 2: Модели и методы / под ред. Д.А. Поспелова. М.: Радио и связь, 1990.

91. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. СПб.: Питер, 2001.

92. Андрейчиков А.В. Интеллектуальные информационные системы: учебник / А.В. Андрейчиков, О.Н. Андрейчикова. М.: Финансы и статистика, 2004. 424 с.

93. Таратухин В.В. Контроль ошибок при виртуальном проектировании сложных технических систем / В.В. Таратухин // Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и систем искусственного интеллекта: науч.-техн. конф. Вологда: ВоГТУ, 2001. С. 121–124.

94. Поспелов Д.А. Многоагентные системы – настоящее и будущее / Д.А. Поспелов // Информационные технологии и вычислительные системы. 1998. № 1.

95. Энциклопедический словарь Ф.А. Брокгауза и И.А. Ефрона. СПб.: Полрадис,1993. 480 с.

96. CASE Data Interchange Format: Integrated Meta-Model – Business Process Modeling Subject Area, CDIF-Draft-BPM-V02, EIA, 1996, 153 pp.

97. Калянов Г.Н. CASE структурный системный анализ (автоматизация и применение) / Г.Н. Калянов. М.: Лори, 1996. 242 с.

98. Калянов Г.Н. Консалтинг при автоматизации предприятий: Научно-практическое издание / Г.Н. Калянов. М.: СИНТЕГ, 1997. 316 с.

99. Смирнова Г.Н. Проектирование экономических информационных систем: учебник / Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов. М.: Финансы и статистика, 2001. 512 с.

100. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник. 2-е изд. / А.М. Вендров. М.: Финансы и статистика, 2005. 544 с.

101. Маклаков С.В. BPwin и Erwin. CASE – средства разработки информационных систем / С.В. Маклаков. М.: ДИАЛОГ-МИФИ, 1999. 256 с.

102. Описание CASE-средств. Режим доступа:www.interface.ru

103. UML – The Unified Modeling Language. Режим доступа: http://www.uml.org.

104. Коналлен Джим. Разработка Web-приложений с использованием UML/ Джим Коналлен. М.: Издательский дом «Вильямс», 2001. 288 с.

105. Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML: пер. с англ. / Л.А. Мацяшек. М.: Издательский дом «Вильямс», 2002. 432 с.

106. Andrei V. Borshchev. Java Engine for UML Based Hybrid State Machines / Andrei V. Borshchev, Yuri B. Kolesov, Yuri B. Senichenkov / 2000 Winter Simulation Conference (WSC'00), December 10–13, 2000, Orlando, Florida, USA Режим доступа: http://www.xjtek.com/files/papers/javaengine2000.pdf.

107. Andrei Borshchev AnyLogic 4.0: Simulating Hybrid Systems with Extended UML-RT / Andrei Borshchev // Simulation News Europe, № 31 April 2001, pp 15-16 Режим доступа: http://www.xjtek.com/files/papers/hybridumlsne2001.pdf.

108. Бенькович Е.С. Практическое моделирование динамических систем / Е.С. Бенькович, Ю.Б. Колесов, Ю.Б. Сениченков. СПб.: БХВ-Петербург, 2002. 464 с.

109. Ойхман Е.Г. Реинжиниринг бизнеса: Реинжиниринг организаций и информационные технологии / Е.Г. Ойхман, Э.В. Попов. М.: Финансы и статистика, 1997. 336 с.

110. Конюх В.Л. Методы имитационного моделирования дискретных систем. Обзор программных продуктов / В.Л. Конюх, Я.Б. Игнатьев, В.В. Зиновьев. Кемерово: Сибирское отделение КНЦ РАН, 2003.

111. Borshchev A. AnyLogic – Multi-Paradigm Simulation for Business, Engineering and Research / A. Borshchev, A. Filippov // The 6th IIE Annual Simulation Solutions Conference, 2004, Orlando, USA Режим доступа: http://www.xjtek.com/files/papers/multiparadigm presentation simsol2004.pdf.

112. Борщёв А.В. От системной динамики и традиционного ИМ – к практическим агентным моделям: причины, технология, инструменты / А.В. Борщёв. Режим доступа: http://www.xjtek.com/files/papers.

113. Christian Wartha. Decision Support Tool – Supply Chain/ Christian Wartha, Momtchil Peev, Andrei Borshchev, Alexei Filippov // 2002 Winter Simulation Conference (WSC'02), December 8–11, 2002, San Diego, California, USA http://www.xjtek.com/files/papers/decisionsupportsupplychain2002.pdf.

114. Borshchev A. Distributed simulation of hybrid systems with AnyLogic and HLA /Andrei Borshchev, Yuri Karpov, Vladimir Kharitonov / Future Generation Computer Systems Volume 18, Issue 6 (May 2002). Режим доступа: http://www.xjtek.com/files/papers/distributedhybridhla2002.pdf.

115. Filippov A. Modeling S-Class Car Seat Control with AnyLogic – Daimler-Chrysler Modeling Contest Object-Oriented Modeling of Embedded Real-Time Systems (OMER-2) / A. Filippov, A. Borshchev. May 2001, Herrsching am Amersee, Germany, pp. 46–50. Режим доступа: http://www.xjtek.com/files.

116. Metcalf S. Spatial Dynamics of Social Network Evolution / Sara Metcalf and Mark Paich // The 23rd International Conference of the System Dynamics Society. July 17 – 21, 2005, Boston, USA. Режим доступа: http://www.xjtek.com/files/papers/spatialdynamics2005.pdf.

117. Christopher Glazner. Optimizing Freight Transportation Policies for Sustainability / A Simulation-Based Investigation of Freight Transportation Policy Planning and Supply Chains // Christopher Glazner

and Sgouris Sgouridis. MIT Sloan School of Management, 2005. Режим доступа: http://www.xjtek.com/files/papers/freighttransportation 2005.pdf.

118. Аксенов К.А. Развитие аппарата прерываний процессов преобразования ресурсов / К.А. Аксенов // На передовых рубежах науки и инженерного творчества: труды третьей международной научно-практической конференции Регионального Уральского отделения Академии инженерных наук им. А.М. Прохорова // Вестник УГТУ-УПИ. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2004. № 15 (45). Ч. 1. С. 187–190.

119. Джексон Питер. Введение в экспертные системы.: пер. с англ.: уч. пос. / Питер Джексон. М.: Издательский дом «Вильямс», 2001. 624 с.

120. Тарасов В. Б. От ИИ к ИЖ: новые направления в науках об искусственном / В.Б. Тарасов // Новости искусственного интеллекта. 1995. № 4. С. 93-117.

121. Тарасов В.Б. Новые стратегии реорганизации и автоматизации предприятий: на пути к интеллектуальным предприятиям / В.Б. Тарасов // Новости искусственного интеллекта. 1996. № 4. С. 40-84.

122. Mark Greaves. Dependable Agent Systems/ MarkGreaves. VictoriaStavridou-Coleman, Robert Laddaga. Intelligent Systems Volume 19, September/October 2004. pp. 20-23.

123. Штейнбух К. Автоматичеловек: кибернетическиефактыигипотезы: пер. снем. / К. Штейнбух; под ред. В.И. Мудрова. М.: Советское радио, 1967. 494 с.

124. Математика и кибернетика в экономике: словарь-справочник / под. ред. Н.П. Федоренко. М.: Экономика, 1975. 700 с.

125. Флейшман Б.С. Элементы теории потенциальной эффективности сложных систем / Б.С. Флейшман. М.: Сов. радио, 1971. 225 с.

126. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии / Г.С. Осипов. М.: Наука; Физматлит, 1997. 112 с.

127. Шенк Р. Обработка концептуальной информации: пер. с англ. / Р. Шенк. М.: Энергия, 1980. 360 с.

128. Kangassalo H. Frameworks of Information Modelling: Construction of Concepts and Knowledge by Using the Intensional Approach: Information Systems Engineering. State of the Art and Research Themes / H. Kangassalo; Ed. by S. Brinkkemper, E. Lindencrona, A. Solberg. – London : Springer, 2000. P. 237-248.

129. Sowa J.F. Conceptual graphs for a database interface / J.F. Sowa // IBM Journal of Research and Development. 1976. № 20:4. P. 336-357.

130. Sowa J.F. Conceptual Structures: Information Processing in Mind and Machine / J. F. Sowa. – Reading, MA: Addison – Wesley, 1984. 481 p.

131. Sowa J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations / J.F. Sowa. – Pacific Grove, CA: Brooks/Cole Publishing Co., 2000. 594 p.

132. Кесс Ю.Ю. Анализ и синтез фреймовых моделейАСУ / Ю.Ю. Кесс. М.: Энергоатомиздат, 1986. 168 с.

133. Словарь по кибернетике / подред. В.С. Михалевича. Киев: Гл. ред. УСЭ им. М.П. Бажана, 1989. 751 с.

134. Математическая энциклопедия: Т. 4 / гл. ред. И.М. Виноградов. М.: Сов. энцикл., 1984. 1216 с.

135. Аксенов К.А. Руководство пользователя системы динамического моделирования ситуаций BPsim2 / К.А. Аксенов. Екатеринбург, 2006. 34 с.

136. Аксенов К.А. Руководство пользователя системы имитационного моделирования BPsim / К.А. Аксенов. Екатеринбург, 2003. 58 с.

137. Марка Д. SADT – Методология структурного анализа и проектирования: пер. с англ. / Д. Марка, К. МакГоуэн. М.: Метатехнология, 1993. 465 с.

138. Суханов В.И. Информационная технология принятия решений при открытой разработке месторождений: дис. ... д-ра техн. наук: 25.00.35 / В.И. Суханов / Ин-т. горного дела УрО РАН. Екатеринбург, 2002. 211 с.

139. Пищулов В. Введение в теорию производства: учеб. пособие / В. Пищулов, К. Рихтер, Е. Дятел. Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2003. 161 с.

140. Городецкий В.И. Многоагентные системы: современное состояние исследований и перспективы применения / В.И. Городецкий // Журнал «Новости искусственного интеллекта». М., 1996. № 1. С. 44–59.

141. Аксенов К.А. Применение к системе сбалансированных показателей аппарата экспертных систем / К.А. Аксенов, Б.И. Клебанов, Н.В. Гончарова // Научные труды международной научно-практической конференции «СВЯЗЬ-ПРОМ 2005» в рамках 2-го Евро-Азиатского международного форума «СВЯЗЬ-ПРОМЭКСПО 2005». Екатеринбург: ЗАО «Компания Реал-Медиа», 2005. С. 69-73.

142. Создание оболочки фреймовой экспертной системы на основе промышленной СУБД: Спецвыпуск / К.А. Аксенов, Н.В. Гончарова, Е.Ф. Смолий, М.Н. Кардаполов, С.И. Печерский, А.А. Плотников, О.М. Смык, И.Ю. Калташев, Б.У. Бобожонов // Вестник УГТУ-УПИ. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2005. С.106–118.

143. Aksyonov K.A. Development of Resource Conversion Processes Model and Simulation System / K.A. Aksyonov, E.F. Smoliy, N.V. Goncharova, A.A. Khrenov, A.A. Baronikhina // Proceedings of the EUROCON 2005. Serbia&Montenegro, Belgrad. 2005. P. 1722–1725.

144. Конфликтные ситуации в интеллектуальных мультиагентных процессах преобразования ресурсов / К.А. Аксёнов, А.С. Зраенко, Н.В. Гончарова, С.Ю. Долматов // Научные труды международной научно-практической конференции «СВЯЗЬ-ПРОМ 2006» в рамках III Евро-Азиатского международного форума «СВЯЗЬ-ПРОМЭКСПО 2006». Екатеринбург: ЗАО «Компания Реал-Медиа», 2006. С. 49-51.

145. Свидетельство об официальной регистрации программы для ЭВМ № 2006611404 «Конструктор фреймовых систем» / К.А. Аксенов, Н.В. Гончарова, Е.Ф. Смолий, М.Н. Кардаполов. Заявл. 2005613159 от 01.12.2005; опубл. 25.04.2006.

146. Wooldridge M., Jennings N., Kinny D. The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems 3. 2000. p. 285-312.

147. Бугайченко Д.Ю. Абстрактная архитектура интеллектуального агента и методы ее реализации. / Д.Ю. Бугайченко, И.П. Соловьев // Системное программирование. – 2005. – № 1. с. 36-67.

148. Маслобоев А.В. Гибридная архитектура интеллектуального агента с имитационным аппаратом / А.В. Маслобоев // Вестник МГТУ – труды Мурманского государственного технического университета.– 2009. – № 1 / Том. 12. вып. 1. – Мурманск: МГТУ, 2009 – С. 113-124.

149. Рыбина Г.В., Паронджанов С.С. Моделирование процессов взаимодействия интеллектуальных агентов в многоагентных системах. / Г.В. Рыбина, С.С. Паронджанов // Искусственный интеллект и принятие решений. 2008. № 3.

150. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. – М.: «АНВИК», 1998. 427 с.

151. Buravova N.A., Aksyonov, K.A., Aksyonova, O.P., Spiricheva, N.R. Modeling the project management process in the teddy designer of the

BPsim.DSS simulation system. International Conference on Numerical Analysis and Applied Mathematics 2018, ICNAAM 2018; Sheraton Rhodes ResortRhodes; Greece; 13-18 September 2018; Code149843. AIP Conference Proceedings. 2019. Volume 2116, 24 July 2019, Number of paper 390005. DOI: 10.1063/1.5114400.

152. K. Aksyonov, H. Ayvazyan, O. Aksyonova, V. Kanev. Development of multi-agent approach and «BPsim.DSS» planning system of oil products supply for gas stations network. 2nd European Conference on Electrical Engineering and Computer Science (EECS 2018). Bern, Switzerland. 20–22 December. DOI: 10.1109/EECS.2018.00015Page(s). C. 33–36.

153. Antonova A., Aksyonov K. Frame-based expert system implementation for resource conversion processes analysis. 4th International Workshop on Radio Electronics and Information Technologies, REIT-Autumn 2018; Yekaterinburg; Russian Federation; 16 November 2018; Код 143170. CEUR Workshop Proceedings Volume 2274, 2018, Pages 23–32.

154. Aksyonov K., Bykov E., Aksyonova O. Petrol delivery management with BPsim.DSS // 33rd Chinese Control Conference, CCC 2014; Nanjing; China; 28 July 2014 through 30 July 2014, Pages 7628–7632. DOI: 10.1109/ChiCC.2014.6896121.

155. Aksyonov K.A., Bykov E.A., Aksyonova O.P. Application of Multi-agent Simulation for Decision Support in a Construction Corporation and its Comparison with Critical Path Method // Applied Mechanics and Materials Vols. 278–280 (2013) Trans Tech Publications, Switzerland. WOS: 000320625001137 doi:10.4028/www.scientific.net/AMM.278–280.2244. pp. 2244–2247.

156. Antonova, A.S., Aksyonov, K.A., Aksyonova, O.P., Kai, W.Analysis of cranes control processes for converter production based on simulation. 1st International Workshop on Radio Electronics and Information Technologies, REIT 2017; Yekaterinburg; Russian Federation; 15 March 2017; Code 126934.CEUR Workshop Proceedings 1814, pp. 21–27.

157. Aksyonov K., Antonova A., Goncharova N. (2018) Analysis of the Electric Arc Furnace Workshop Logistic Processes Using Multiagent Simulation. In: Thampi S.M., Krishnan S., Corchado Rodriguez J.M., Das S., Wozniak M., Al-Jumeily D. (eds) Advances in Signal Processing and Intelligent Recognition Systems. SIRS 2017. Advances in Intelligent Systems and Computing, vol 678. Springer, Cham. DOI https://doi.org/10.1007/978-3–319-67934–1_35. P. 390-397. WOS: 000449761600035.

158. Aksyonov K., Antonova A., Wang Kai, Aksyonova O. Rules for construction of simulation models for production processes optimization.

3rd International Workshop on Radio Electronics and Information Technologies, REIT-Spring 2018; Institute of Radioelectronics and Information Technologies of Ural Federal UniversityYekaterinburg; Russian Federation; 14 March 2018; Code 135280. CEUR Workshop Proceedings Vol. 2076, 2018, P. 9–18.

159. Aksyonov K., Antonova A. Application of a metallurgical enterprise information system for collection and analysis of big data and optimization of multi-agent resource conversion processes. 2018 International Research Workshop on Information Technologies and Mathematical Modeling for Efficient Development of Arctic Zone, IT and Math AZ 2018; Graduate School of Business and ManagementYekaterinburg; Russian Federation; 19 April 2018 до 21 April 2018; Код 137083. CEUR Workshop Proceedings. Volume 2109, 2018, P. 1–6.

160. Aksyonov K., Antonova A. Development of a Hybrid Decision-Making Method Based on a Simulation-Genetic Algorithm in a Web- Oriented Metallurgical Enterprise Information System. The tenth International Conference on Ubiquitous and Future Networks (ICUFN 2018). 3–6 July 2018, Prague, Czech Republic. Page(s): 197-202.

161. Aksyonov, K., Antonova, A. The use of simulation in the management of converter production logistics processes. Advances in Intelligent Systems and Computing. International Symposium on Intelligent Systems Technologies and Applications, ISTA 2016; Jaipur; India; 21 September 2016 through 24 September 2016; Volume 530, pp. 675–682. DOI: 10.1007/978-3–319–47952–1_54.

162. Aksyonov K., Bykov E., Aksyonova O., Goncharova N., Nevolina A. Perspectives of Modeling in Metallurgical Production (WIP) // Society for Modeling & Simulation International (SCS). 2015 Summer Simulation Multi-Conference (SummerSim'15). Chicago. USA. 26–29 July, 2015. Simulation Series. Volume 47, Issue 10, 2015, P. 341-344.

163. Aksyonov K., Bykov E., Aksyonova O., Goncharova N., Nevolina A. Analysis of Simulation Modeling Systems Illustrated with the Problem of Model Design for the Subject of Technological Logistics (WIP) // Society for Modeling & Simulation International (SCS). 2015 Summer Simulation Multi-Conference (SummerSim'15). Chicago. USA. 26–29 July, 2015. Simulation Series. Volume 47, Issue 10, 2015, P. 345-348.

164. Aksyonov K.A., Antonova A.S., Klebanov B.I., Panteleeva Y.S., Krymov Zh.A. Examples of solving optimization problems of metallurgical manufacture based on using simulation system Plant Simulation from

Siemens // 2nd International Conference on Advanced Education Technology and Management Science (AETMS). Hong Kong, China, Dec. 25-26, 2014, Pages: 372–376. WOS:000352704700070.

165. Aksyonov K., Bykov E., Aksyonova O. Real time simulation models integrated into the corporate information systems // 33rd Chinese Control Conference, CCC 2014; Nanjing; China; 28 July 2014 through 30 July 2014, Pages 6810-6813.WOS:000366482806175, DOI: 10.1109/ChiCC.2014.6896121.

166. Aksyonov K.A., Spitsina I.A., Sysoletin E.G., Aksyonova O.P., Smoliy E.F. Multi-agent approach for the metallurgical enterprise information system development // 24th Int. Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo'2014). 7–13 September, Sevastopol. Vol. 1. P. 437-438. DOI: 10.1109/CRMICO.2014.6959467.

167. Medvedev S.N., Aksyonov K.A. Comparative analysis of multiagent systems and Kanban system in the construction of operational production plans for engineering enterprises // 24th Int. Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo'2014). 7–13 September, Sevastopol. Vol. 1. P. 433–434. DOI: 10.1109/CRMICO.2014.6959465.

168. Aksyonov K., Bykov E., Aksyonova O., Goncharova N., Nevolina A., Real-time simulation modeling of logistics in metallurgical production, Proceedings of the 5th IASTED International Conference on Modelling, Simulation and Identification (MSI 2014), July 16–18, 2014, Banff, Canada, pp. 30-37. DOI: 10.2316/P.2014.820-025.

169. Аксенов К.А., Клебанов Б.И., Белан С.Б., Перескоков С.А. Реализация типовых бизнес процессов управления качеством изделий с использованием модулей автоматизированной системы выпуска металлургической продукции // Современные наукоемкие технологии. – 2018. – № 12 (часть 1). URL: http://www.top-technologies.ru/ru/article/view?id = 37254 (дата обращения: 29.01.2019).

170. Беренов Д.А., Белан С.Б., Аксенов К.А., Перескоков С.А. Полностью оцифрованное металлургическое производство: слежение, аналитика, моделирование // Фундаментальные исследования. – 2017. – № 9-2. – С. 272-277; URL: http://www.fundamental-research.ru/ru/article/view?id = 41739 (дата обращения: 29.05.2018).

171. Чирышев Ю.В., Аксенов К.А., Сысолетин Е.Г., Белан С.Б., Перескоков С.А. Разработка инструментальных средств для испытаний автоматизированной системы выпуска металлургической продукции //

Современные наукоемкие технологии. – 2016. – № 12-2. – С. 294–298; URL: http://www.top-technologies.ru/ru/article/view?id = 36438 (дата обращения: 07.02.2017).

172. Спицина И.А., Белан С.Б., Перескоков С.А., Аксенов К.А., Сысолетин Е.Г. Разработка автоматизированного рабочего места персонала для автоматизированной системы выпуска металлургической продукции // Фундаментальные исследования. – 2016. – № 10-3. – С. 571-575; URL: http://www.fundamental-research.ru/ru/article/view?id = 40897 (дата обращения: 07.02.2017).

173. Аксенов К.А., Кондратьев А.С., Медведев С.Н., Белан С.Б., Перескоков С.А. Разработка принципов мультиагентной имитационной модели, реализующей алгоритмы перецеховки // Фундаментальные исследования. – 2016. – № 9-3. – С. 455-460; URL: http://www.fundamental-research.ru/ru/article/view?id = 40765 (дата обращения: 07.02.2017).

174. Аксенов К.А., Антонова А.С., Чирышев А.В., Медведев С.Н. Разработка конструктора деревьев анализа процессов для автоматизированной системы выпуска металлургической продукции // Фундаментальные исследования. № 9 (часть 2) 2015, С. 209-214. URL: http://fundamental-research.ru/ru/article/view?id = 39075 (дата обращения: 19.01.2016).

175. Аксенов К.А., Антонова А.С., Дудин Н.А., Аксенова О.П., Смолий Е.Ф. Разработка имитационной модели технологической логистики цехов горячей и холодной прокатки // Современные проблемы науки и образования. – 2015. – № 2; URL: http://www.science-education.ru/129-22093 (дата обращения: 09.10.2015).

176. Антонова А.С., Аксенов К.А., Ростунцев С.Д., Рудь А.И. Реализация эволюционно-имитационного алгоритма оптимизации процессов в системе моделирования производственных процессов металлургического предприятия // Современные проблемы науки и образования. – 2015. – № 2; URL: www.science-education.ru/122-21220 (дата обращения: 08.09.2015).

177. Аксенов К.А., Антонова А.С., Айзатуллов А.М. Анализ работы электросталеплавильного цеха с помощью имитационного моделирования // Современные проблемы науки и образования. – 2015. – № 2;URL: www.science-education.ru/122-21153 (дата обращения: 08.09.2015).

178. Аксенов К.А. Принципы построения моделей мультиагентного процесса преобразования ресурсов // Фундаментальные исследования. – 2015. – № 5 (часть 1), май, 2015. С. 14–18. URL: https://www.fundamental-research.ru/ru/article/view?id = 38000 (дата обращения: 27.09.2015).

179. Сысолетин Е.Г., Аксенов К.А., Круглов А.В. Интеграция гетерогенных информационных систем современного промышленного предприятия // Современные проблемы науки и образования. – 2015. – № 1; URL: www.science-education.ru/121–19030 (дата обращения: 25.05.2015).

180. Borodin, A., Kiselev, Y., Mirvoda, S., Porshnev, S., On design of domain-specific query language for the metallurgical industry. Proceedings of the 11th International Conference BDAS: Beyond Databases, Architectures and Structures: Communications in Computer and Information Science, pp. 505–515, 2015.

181. Borodin, A., Mirvoda, S., Kulikov, I., Porshnev, S., Optimization of memory operations in generalized search trees of PostgreSQL. Proceedings of the International Conference: Beyond Databases, Architectures and Structures, pp. 224–232, 2017.

ПРИЛОЖЕНИЯ. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СДМС

Приложение 1

Выбор критериев сравнения

Сравнение функциональных возможностей систем AnyLogic 5.2, ARIS 5.0, G2 и BPsim2, при построении мультиагентных моделей процессов преобразования ресурсов, рассмотрим на примере построения модели базового предприятия, описанного в разделе 4.1.

Для сравнительной оценки функциональных возможностей пакетов рассмотрим их при описании:

1) проектирования концептуальной модели предметной области;

2) модели предприятия в целом:

- иерархической структуры модели процесса преобразования ресурсов;
- элементов слияния и разветвления процессов;
- функционально-стоимостного анализа модели процесса;
- динамического анализа данных, анализа результатов во время эксперимента;
- описания графов целей и поддержки методики BSC (цели, страты, причинно-следственные связи);
- поддержки русского языка;
- изменения параметров модели во время эксперимента.

3) процесса выполнения заказов на производстве:

- векторов входов / выходов / средств;
- объема захватываемого / освобождаемого ресурса / средства – постоянная / функция / случайная величина;
- длительности операции: постоянная / функция / случайная величина;
- приоритета операции и типа приоритета;
- условия запуска элемента преобразования – функция времени и ресурсов, и средств;
- модели ресурса и средства;
- модели интеллектуального агента (база знаний (правила диагностирования ситуаций, решения/действия), цели агента, сообщения);

4) расчета параметров «штрафные санкции» и «стоимость заказа».

5) при наличии специальных возможностей описания целей модели:

- в виде графа;
- в виде карты BSC.

6) модели на ограниченном естественном языке;

7) при наличии средств описания команд;

8) возможности построения мультиагентных моделей. Наличие агентов (моделей ЛПР), обладающих индивидуальным поведением и знаниями:

- наличие класса объекта «агент», на основе которого можно создавать экземпляры класса;
- возможность описания поведения агента (язык описания сценариев поведения агентов (диагностирование ситуаций, поиск решения), язык описания команд управления);
- язык описания знаний агента;
- язык обмена сообщениями между агентами;
- удобство интерфейса пакета при описании агентов.

9) поддержки ИМ;

10) поддержки аппарата ЭС:

- база знаний;
- механизм логического вывода из базы знаний.

11) поддержки ситуационного подхода (наличие механизмов описания ситуаций);

12) экспорта/ импорта/ открытого доступа к базе знаний модели:

- импорта начальных условий и исходных данных из внешних источников;
- экспорта результатов экспериментов;
- открытого доступа к базе знаний модели.

При проведении сравнительного анализа мультиагентных систем моделирования возможность представления элемента процесса преобразования ресурсов рассмотрим на примере процесса выполнения заказов на производстве; расчет параметра (характеристики) рассмотрим на примере вычисления штрафных санкций и стоимости заказа.

Также уделим внимание следующим критериям анализа:

- удобство интерфейса пакета при описании элементов мультиагентного процесса преобразования ресурсов.

В табл. П.1 приведены описания моделей агентов базового предприятия.

Таблица П.1

Описание агентов, их целей и базы знаний

Агент	Цели	База знаний
1	2	3

1А – Потребитель (внешний объект)	1. $S_i \rightarrow \min S_i\{n,k,t,s\}$, т. е. минимизировать стоимость заказа
	2. $T_i \rightarrow \min T_i\{t,k,n\}$, т. е. минимизировать время заказа
		If (1А в состоянии «потенциальный потребитель») $\wedge$ (моделируется воздействие рекламы, с заданной эффективностью Ad_Effectiveness по таймауту, подчиняющемуся экспоненциальной зависимости от модельного времени) $\wedge$ (моделируется передача информации во время контактов агента – «потребителя» с другими потребителями – задается среднее количество контактов в день Contact_Rate) $\wedge$ (задается сила убеждения агента – «потребителя» – в виде параметра агента Adoption_Fraction) Then (1А переходит в состояние «потребитель») (0)
		
		If (1А в состоянии «потребитель», сформирована потребность в товарах или услугах) Then (менеджеру 2А – заказ продукции количество N, срок выполнения T, предельное значение стоимости S^*, предельное значение количества дней задержки выполнения заказа U^*) $\wedge$ ($r_i = 0$) (0)
		
		If (от менеджера 2А получено сообщение о подписании договора, время задержки $Ui \le U^*$ стоимость заказа $S_i \le S^*$) Then (бухгалтеру 3А перечисление денежных средств Si) (2)
		
		If (от менеджера 2А получено сообщение, что заказ Zi выполнен) $\wedge$ ($T_i^* \le T_i$) Then (принять заказ) (0)
		If (от менеджера 2А получено сообщение, что заказ Zi выполнен) $\wedge$ ($T_i^* > T_i$) Then (принять заказ) (0) $\wedge$ (получить штрафные санкции $H_i = 0{,}1\,\%\ \tau_i S_i$) (2)
		
		If (1А в состоянии «потребитель » уже сделал заказ) $\wedge$ (сработал таймаут, определенный параметром Discard_Time) Then (1А переходит в состояние «потенциальный потребитель»)

1	2	3

2А – Менеджер	1. $N_i \rightarrow \max\ N_i$, т. е. максимизировать количество заказов
	2. Сохранение и расширение клиентской базы предприятия и доли рынка
	3. $S_i \rightarrow \max\ S_i\{n,k,t,s\}$, т. е. максимизировать стоимость заказа
		If (от потребителя 1А поступило сообщение о заказе продукции N, срок выполнения T) Then (увеличить счетчик заказов $i = i + 1$) (0) $\wedge$ (открыть заказ Zi продукции $N_i = k_i n$, срок выполнения T_i)(0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \le 10n$) $\wedge$ ($T_i \ge t$) Then (расчет стоимости заказа $S_i = sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = p * k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = k_i = N_i/n$)(0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \le 10n$) $\wedge$ ($t > T_i \ge t/2$) Then (расчет стоимости заказа $S_i = sk_i + 20\%sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 2p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 2k_i = 2N_i/n$) (0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \le 10n$) $\wedge$ ($t/2 > T_i \ge t/4$) Then (расчет стоимости заказа $S_i = sk_i + 40\%sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 4p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 4k_i = 4N_i/n$) (0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($10n \le N_i < 20n$) $\wedge$ ($T_i \ge t$) Then (расчет стоимости заказа $S_i = sk_i - 7\%sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = p * k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = k_i = N_i/n$)(0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($10n \le N_i < 20n$) $\wedge$ ($t > T_i \ge t/2$) Then (расчет стоимости заказа $S_i = (sk_i - 7\%sk_i) + 20\%(sk_i - 7\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 2p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 2k_i = 2N_i/n$)(0)
		
		If (заказ Zi продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($10n \le N_i < 20n$) $\wedge$ ($t/2 > T_i \ge t/4$) Then (расчет стоимости заказа $S_i = (sk_i - 7\%sk_i) + 40\%(sk_i - 7\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 4p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 4k_i = 4N_i/n$)(0)

Продолжение табл. П.1

1	2	3

		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($20n \le N_i < 50n$) $\wedge$ ($T_i \ge t$) Then (расчет стоимости заказа $S_i = sk_i - 10\%sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = p * k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = k_i = N_i/n$)(0)
		
		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($20n \le N_i < 50n$) $\wedge$ ($t > T_i \ge t/2$) Then (расчет стоимости заказа $S_i = (sk_i - 10\%sk_i) + 20\%(sk_i - 10\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 2p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = = 2k_i = 2N_i/n$)(0)
		
		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($20n \le N_i < 50n$) $\wedge$ ($t/2 > T_i \ge t/4$) Then (расчет стоимости заказа $S_i = (sk_i - 10\%sk_i) + 40\%(sk_i - 10\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 4p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 4k_i = 4N_i/n$)(0)
		
		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \ge 50n$) $\wedge$ ($T_i \ge t$) Then (расчет стоимости заказа $S_i = sk_i - 15\%sk_i$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = p * k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = k_i = N_i/n$)(0)
		
		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \ge 50n$) $\wedge$($t > T_i \ge t/2$) Then (расчет стоимости заказа $S_i = (sk_i - 15\%sk_i) + 20\%(sk_i - 15\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 2p*k_i$) (0)$\wedge$ (расчет необходимого кол-ва рабочих $x_i = = 2k_i = 2N_i/n$)(0)
		
		If (заказ *Zi* продукции $N_i = k_i n$, срок выполнения T_i) $\wedge$ ($N_i \ge 50n$) $\wedge$ ($t/2 > T_i \ge t/4$) Then (расчет стоимости заказа $S_i = (sk_i - 15\%sk_i) + 40\%(sk_i - 15\%sk_i)$) (0) $\wedge$ (расчет необходимых производственных мощностей $p_i = 4p*k_i$) (0) $\wedge$ (расчет необходимого кол-ва рабочих $x_i = 4k_i = 4N_i/n$)(0)

1	2	3

		If (расчет необходимых производственных мощностей p_i выполнен) $\wedge$ (расчет необходимого кол-ва рабочих x_i выполнен) $\wedge$ (стоимость S_i рассчитана) Then (расчет времени задержки выполнения заказа U_i по массиву плановой загрузки производственных мощностей B: $U_i = 0$, цикл пока $((ST - B[\Theta + \mu + U_i]) < p_i)$, выполнить $(U_i = U_i + 1))(0) \wedge (r_i = 4)(0)$
		
		If ($r_i = 4$) $\wedge$ (время задержки $U_i > U^*$ или стоимость заказа $S_i > S^*$, т. е. больше предельных значений потребителя 1А) Then (от потребителя 1А получен отказ по заказу Zi) $\wedge$ ($r_i = 3$) (0) $\wedge$ (сообщение директору 6А об отказе потребителя 1А) (0)
		If ($r_i = 4$) $\wedge$ (время задержки $Ui \le U^*$) $\wedge$ (стоимость заказа $S_i \le S^*$) Then (подписание договора об открытии заказа с потребителем 1А) (0) $\wedge$ ($r_i = 1$) (0) $\wedge$ (сообщение бухгалтеру 3А о том, что должны быть перечислены денежные средства Si от потребителя 1А) (0)
		
		If (с потребителем 1А подписан договор об открытии заказа) $\wedge$ ($r_i = 1$) Then (формирование заявки на производство заказа Zi для начальника производства 4А, кол-во продукции Ni и срок выполнения Ti) (0) $\wedge$ (формирование заявки на ресурсы заказа Zi для снабженца 5А, кол-во продукции Ni и срок выполнения Ti) (0) $\wedge$ (формирование заявки бухгалтеру 3А на резервирование денежных средств на проплату за ресурсы в размере 0,1 S_i) (0)
		
		If (от начальника производства 4А получено сообщение, что произведено по заказу Zi продукции N_i за время $T_i^*) \wedge (T^*_i > T_i)$ Then (рассчитать сумму штрафных санкций $H_i = 0{,}1\,\% \ (T^*_i - T_i)S_i)$ (0) $\wedge$ (передача сообщения H_i бухгалтеру 3А) (0) $\wedge$ (передача отчета директору 6А) (0) $\wedge$ (подготовка документов на отгрузку продукции по заказу Zi потребителю 1А) (0)
		
		If (от начальника производства 4А получено сообщение, что произведено по заказу Zi продукции N_i за время $T_i^*) \wedge (T^*_i \le T_i)$ Then (штрафные санкции $H_i = 0)(0) \wedge$ (передача сообщения бухгалтеру 3А) (0) $\wedge$ (передача отчета директору 6А) (0) $\wedge$ (подготовка документов на отгрузку продукции по заказу Zi потребителю 1А) (0)

Продолжение табл. П.1

1	2	3

3А – Бухгалтер	1. $S_i \rightarrow \max S_i\{n,k,t,s\}$, т. е. максимизировать стоимость заказа
	2. Максимизировать рентабельность, чистую прибыль
		If (от менеджера 2А получено сообщение об открытии заказа *Zi*, стоимость *Si*) $\wedge$ (на счет предприятия пришли ДС от потребителя 1А в размере S_i) $\wedge$ ($r_i = 1$) Then (сообщение снабженцу 5А о получении ДС по заказу *Zi*)(0)
		
		If (от менеджера 2А поступила заявка на резервирование денежных средств на проплату за ресурсы по заказу *Zi*) Then (резервирование суммы ДС в размере 0,1 S_i) (0)
		
		If (от снабженца 5А поступили счета на проплату МТР по заказу *Zi*, сумма *Si* , ресурсы = 0,1 S_i) Then (проплата ДС по счетам) (2)
		
		If (от начальника производства 4А поступили счета на оплату поиска дополнительных рабочих по заказу *Zi* на сумму $\Delta \cdot y_i$) Then (оплата ДС по счетам) (2)
		
		If (счета по заказу *Zi* проплачены) Then (сообщение снабженцу 5А) (0)
		
		If ($r_i = 2$) $\wedge$ (от менеджера 2А сообщение, что прошла отгрузка по заказу *Zi* потребителю 1А) $\wedge$ (штрафные санкции *Hi* > 0) Then (перечисление потребителю 1А суммы H_i) (2)
		
		If ($r_i = 2$) $\wedge$ (от менеджера 2А сообщение, что прошла отгрузка по заказу *Zi* потребителю 1А) $\wedge$ (предоставлены на оплату наряды на зарплату рабочих x_i, y_i от начальника производства 4А) Then (расчет прямых затрат *Si* пр = *Si* ресурсы + 0,2 S_i + + $\Delta \cdot y_i$) (0) $\wedge$ (расчет накладных расходов размере 0,5 *Si*) (2) $\wedge$ (расчет прибыли от пр-ва) (0)
		
		If (закончился отчетный период) Then (расчеты с персоналом по з/п) (0) $\wedge$ (расчет рентабельности) (1) $\wedge$ (расчет и уплата налогов) (3) $\wedge$ (расчет чистой прибыли) (3) $\wedge$ (определение финансового состояния предприятия) (4) $\wedge$ (расчет инвестиций в пр-во) (4) $\wedge$ (расчет с дебиторами и кредиторами) (4) $\wedge$ (передача отчета директору) (5)

1	2	3

4А – Начальник производства	1. $Z \rightarrow \max\ Z\{p\}$, т. е. максимизировать загрузку производственных мощностей и рабочей силы
	2. $t \rightarrow \min$, т. е. уменьшить время на производство партии продукции (увеличить производитель-ность труда)
	3. $b_i \rightarrow \min$, т. е. минимизировать кол-во брака
		If ($r_i = 1$) $\wedge$ (от менеджера 2А поступила информация о заказе z_i, кол-во продукции N_i и срок выполнения T_i) $\wedge$ (поступили материальные ресурсы от снабженца 5А) $\wedge$ ($p^* \geq p_i$, т. е. есть свободные производственные мощности) Then ($\Theta_i = \Theta$ начинается выполнение заказа z_i) (0) $\wedge$ ($p^* = p^* - p_i$) (0) $\wedge$ ($x^* = x^* - x_i$) (0) $\wedge$ (заведение информации по загрузке производственных мощностей в массив плановой загрузки B: $B[\Theta] = B[\Theta] + Pi$, $B[\Theta + 1] = B[\Theta + 1] + Pi$ ... $B[\Theta + Ti] = B[\Theta + Ti] + Pi)(Ti)$
		
		If (поступила информация о заказе z_iот менеджера 2А, кол-во продукции N_i и срок выполнения T_i) $\wedge$ (поступили материальные ресурсы от снабженца 5А) $\wedge$ ($x^* < 0$, т. е. нехватка рабочих) Then (запрос на рынок труда о привлечении $y_i = \lvert x^* \rvert$ рабочих нужных специальностей по заказу z_i) (0) $\wedge$ (запрос бухгалтеру на оплату затрат на поиск y_i доп. Рабочих, которые равны $y_i \cdot \Delta$) (0) $\wedge$ ($x^* = 0$)(0) $\wedge$ ($x_i = x_i - y_i$) (0)
		
		If (привлечено y_i дополнительных рабочих по заказу z_i) Then (организация их работы по заказу z_i) (0)
		
		If (поступила информация о заказе z_i от менеджера 2А, кол-во продукции N_i и срок выполнения T_i) $\wedge$ (поступили материальные ресурсы от снабженца 5А) $\wedge$ ($p^* < p_i$, т. е. нет свободных производственных мощностей) Then (заказ z_iприостанавливается из-за нехватки производственных мощностей на 1 день)(1)
		
		If (по заказу z_i произведено кол-во продукции N_i) Then (контроль качества продукции N_i) (0)
		
		If ($b_i \neq 0$, т. е. часть продукции забракована) Then (выделение рабочих на исправление брака в количестве $[(Xi + Yi)^* b_i / Ni] + 1)$ $\wedge$ (возврат забракованной части z_i на устранение брака на время $[Ti * b_i / Ni] + 1)$ $(Ti * b_i / Ni)$
		
		If ($b_i = 0$, т. е. продукция z_iпрошла контроль качества) Then (сдача z_iна склад готовой продукции) (0) $\wedge$ (упаковка z_i) (1) $\wedge$ (предпродажная подготовка z_i) (0)
		If (устранен брак b_i, т. е. продукция z_i прошла контроль качества) Then (сдача z_i на склад готовой продукции) (0) $\wedge$ (упаковка z_i) (1) $\wedge$ (предпродажная подготовка z_i) (0)

Продолжение табл. П.1

1	2	3
		If (продукция z_i сдана на склад готовой продукции) Then (сообщение про готовность z_i, N_i менеджеру 2А) (0) ∧ (сообщение бухгалтеру 3А x_i, y_i для начисления з/п) (0) ∧ (передача отчета про z_iдиректору) (0) ∧ ($r_i = 2$) (0) ∧ ($T^*_i = \Theta - \Theta_i$) (0) ∧ ($x^* = x^* + x_i$) (0) ∧ ($p^* = p^* + p_i$) (0) ∧ ($y_i$ рабочих уволить) (0) ∧ (сообщить менеджеру 2А рассчитанную T^*_i) (0)

5А – Управляющий поставками (снабженец)	1. $m \rightarrow \min$, т.е. минимизировать долю стоимости заказа на проплату МТР при том же качестве продукции
	2. $tm \rightarrow \min$, т.е. минимизировать время прихода МТР на производство
		If (поступила заявка на ресурсы от менеджера 2А, кол-во продукции N_i и срок выполнения T_i) Then (заказ ресурсов у поставщиков в расчете на k_i партий) (0)
		If (получены счета от поставщиков) Then (передача счетов на проплату бухгалтеру 2А) (0)
		
		If (получено сообщение от бухгалтера 2А о проплате ДС) Then (доставка ресурсов от поставщиков) (3)
		If (ресурсы от поставщика доставлены) Then (входной контроль ресурсов) (0)
		
		If (ресурсы от поставщика прошли входной контроль) Then (выдача ресурсов начальнику производства 4А) (0) ∧ (передача отчета директору) (0)
		
		If (ресурсы от поставщика не прошли входной контроль) Then (возврат ресурсов поставщику) (0) ∧ (передача отчета директору) ∧ (заказ недостающих ресурсов у поставщиков) (0)

6А – Директор	1. Обеспечение инвестиций в производство (обновление оборудования, расширение производства)
	2. Обучение кадрового состава, его профессионализа-ция (инвестиции в персонал)
	3. Максимизация рентабельности, чистой прибыли
		If (получен отчет от бухгалтера 3А) Then (анализ рентабельности) (0) ∧ (анализ состояния расчетов с бюджетами и внебюджетными фондами) (0) ∧ (анализ чистой прибыли) (0) ∧ (анализ финансового состояния предприятия) (0) ∧ (анализ инвестиций в пр-во) (0) ∧ (анализ расчетов с дебиторами и кредиторами) (0) ∧ (выработка корректирующего воздействия) (5)
		
		If ($r_i = 2$) ∧ (получен отчет от бухгалтера 3А) ∧ (прибыль по заказу отрицательна pribi < 0) Then (увеличение счетчика убыточных заказов на 1)(0)
		
		If ($r_i = 2$) ∧ (получен отчет от бухгалтера 3А) ∧ (сумма штрафа Hi > 0) Then (увеличение счетчика заказов со штрафом на 1) (0)

Окончание табл. П.1

1	2	3

	4. Сохранение и расширение клиентской базы предприятия и доли рынка
	5. Обеспечение инвестиционной привлекательности предприятия
		If (r_i = 2) ∧ (получен отчет от бухгалтера 3А) ∧(количество дополнительных рабочих Yi > 0) Then (увеличение счетчика заказов с дополнительными рабочими на 1) (0)
		
		If (r_i = 2) ∧ (счетчик убыточных заказов > = 3) ∧ (были заказы со штрафами за просрочку, т. е. не хватает производственных мощностей) Then (докупить два станка из прибыли предприятия) ∧ (счетчик заказов со штрафом обнулить) (0)
		
		If (r_i = 2) ∧ (счетчик убыточных заказов > = 3) ∧ (были заказы с дополнительными рабочими, т. е. не хватает рабочей силы) Then (принять двоих основных рабочих) ∧ (счетчик заказов с дополнительными рабочими обнулить) (0)
		
		If (r_i = 2) ∧ (получен отчет от бухгалтера 3А) ∧ (прибыль по заказу отрицательна pribi < 0) Then (увеличение нормативной стоимости одной партии заказа на 3 %) (0)
		
		If (r_i = 2) ∧ (получен отчет от бухгалтера 3А) Then (подсчет и анализ чистой прибыли) (0)
		
		If (получен отчет от начальника производства 4А) ∧ (предприятие работает более месяца) Then (анализ средней загрузки производственных мощностей)(0) ∧ (анализ средней загрузки рабочей силы) (0)
		
		If (получен отчет от начальника производства 4А) ∧(предприятие работает более двух месяцев) ∧ (средняя загрузка производственных мощностей менее половины имеющихся) ∧ (более месяца не было корректирующего воздействия) Then (выработка корректирующего воздействия: продать два станка, увеличить прибыль предприятия на вырученные средства) (0)
		
		If (получен отчет от начальника производства 4А) ∧ (предприятие работает более двух месяцев) ∧ (средняя загрузка рабочей силы менее половины имеющихся) ∧ (более месяца не было корректирующего воздействия) Then (выработка корректирующего воздействия: уволить двух рабочих) (0)

Примечание. В скобках указана длительность выполнения данной операции в днях; округление – всегда в большую сторону.

Приложение 2

Система имитационного моделирования AnyLogic5.2

На рис. П.2.1 представлены основные процессы базового предприятия в СИМ AnyLogic.

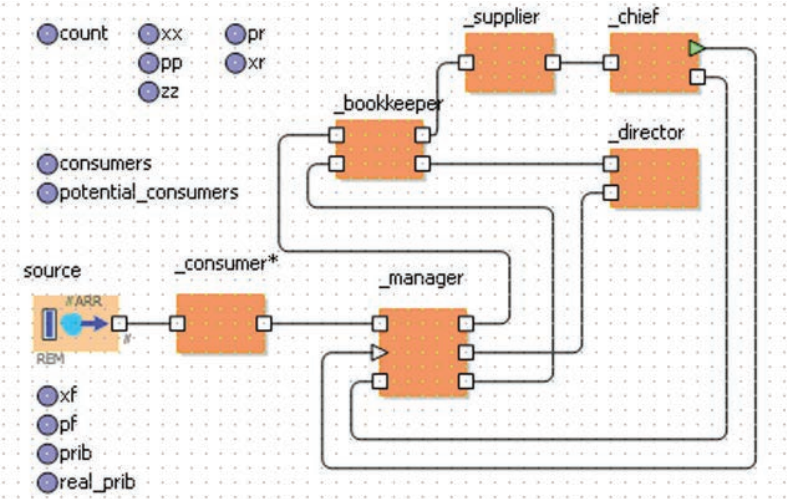


Рис. П.2.1. Модель базового предприятия в СИМ AnyLogic 5.2

Модель элемента процесса преобразования «выполнение заказов на производстве» (рис. П.2.2):

- вектора входов / выходов / средств представлены входами («заказы на выполнение»; выходами («выполненные заказы»); средствами («массив плановой загрузки производственных мощностей», «количество свободных производственных мощностей», «количество свободных рабочих»).
- объем захватываемого / освобождаемого ресурса / средства, описывается на языке Java в виде постоянной / функции / случайной величины. Захват / освобождение необходимого количества ресурсов описывается в виде конструкции на языке Javaследующего вида:

```
if (mess.pi <= ch_m.pf)
{
ch_m.pf = ch_m.pf-mess.pi;
ch_m.xf = ch_m.xf-mess.xi;
}
```

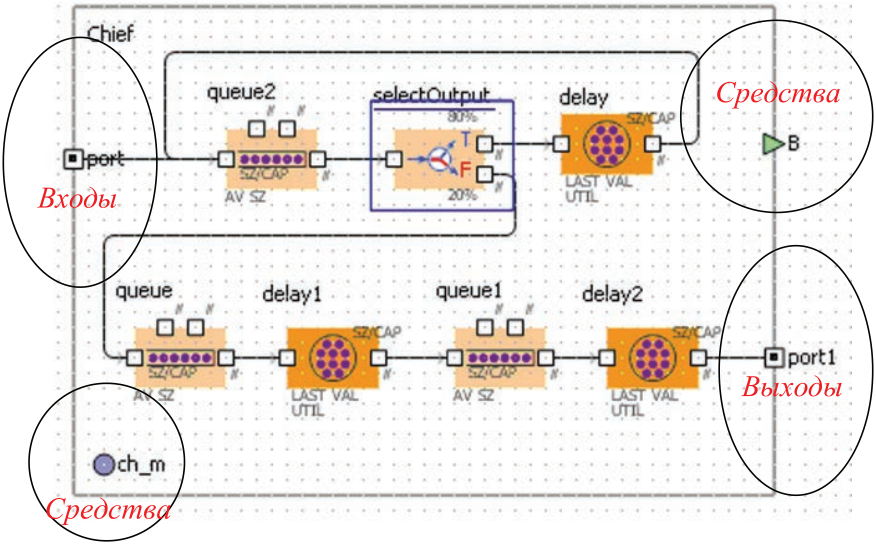


Рис. П.2.2. Вектора входов/выходов/средств

В качестве средств могут быть описаны: исполнители работы, оборудование, аппаратное обеспечение, программное обеспечение, машинный ресурс. С помощью объектов на языке Java можно ввести необходимое количество средств;

• длительность операции: любой объект, описываемый на языке Java, в частности, постоянная / функция / случайная величина. На рис. П.2.3 длительность равна значению переменной *Ti*;

• приоритет операции не устанавливается, а задается тип очереди FIFO (первым пришел – первым ушел), специальными средствами можно создать очередь LIFO (последним пришел – первым ушел) или смоделировать приоритетную очередь. Выбор типа очереди показан на рис. П.2.4;

• условие запуска элемента преобразования ресурсов описывается функцией на языке Java, в которой могут использоваться любые переменные модели. Например, это может быть функция времени, ресурсов или средств. Так, на рис. П.2.5 представлено следующее условие запуска операции «Выполнение заказа»: количество необходимых производственных мощностей должно быть

не больше количества свободных производственных мощностей. Если это условие не выполняется, запуск элемента преобразования откладывается на единицу модельного времени (один день), после чего условие запуска проверяется вновь.

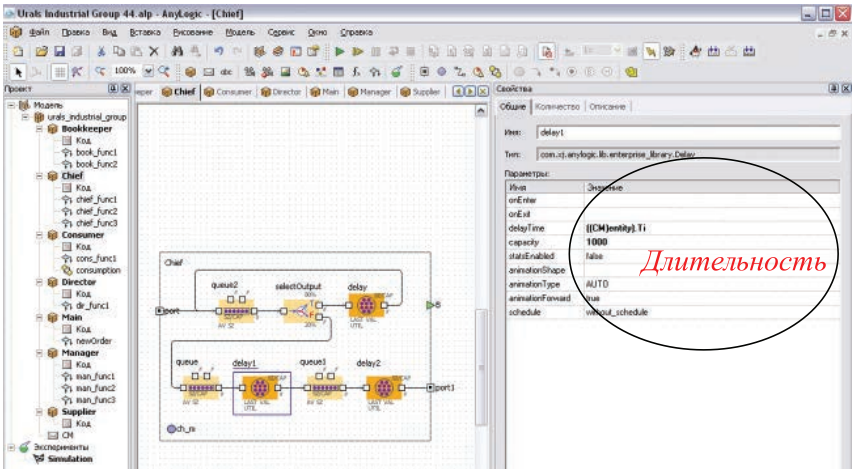


Рис. П.2.3. Длительность операции

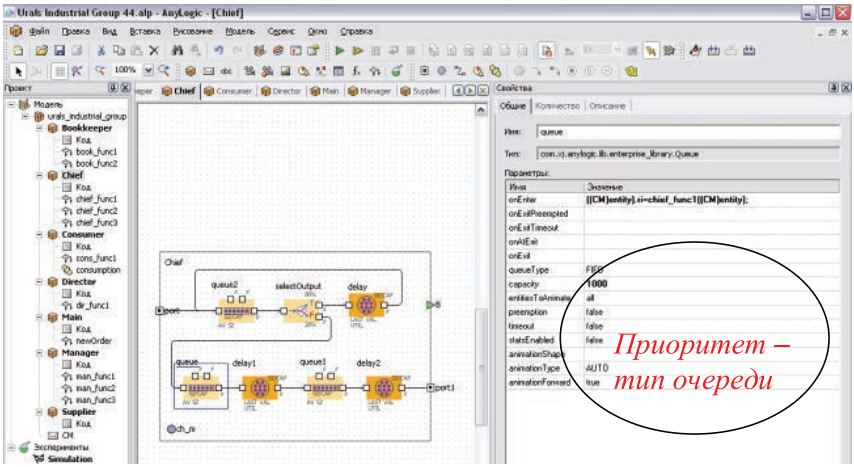


Рис. П.2.4. Приоритет и тип приоритета

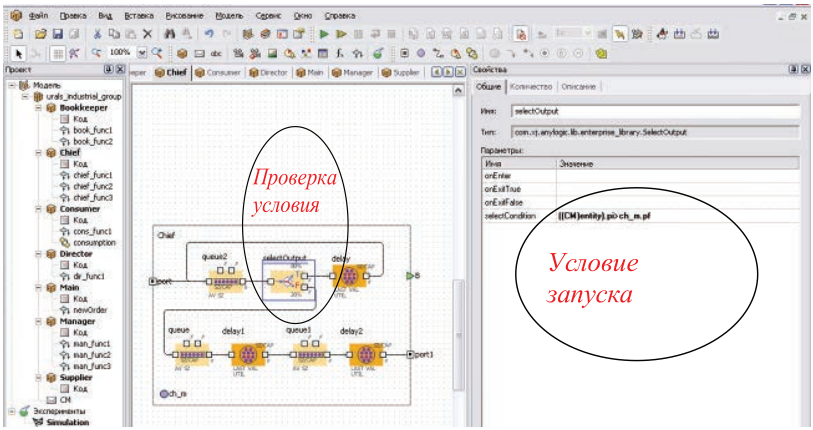


Рис. П.2.5. Условие запуска элемента преобразования

Модель ресурса и средства:

- описывается любым объектом на языке Java;
- функциями на языке Java могут быть описаны ограничения, минимум, максимум и т. д.

Так, на рис. П.2.6 представлено описание количества производственных мощностей ST = 70 и количества трудовых ресурсов PR = 50. Могут быть заданы любые типы ресурсов.

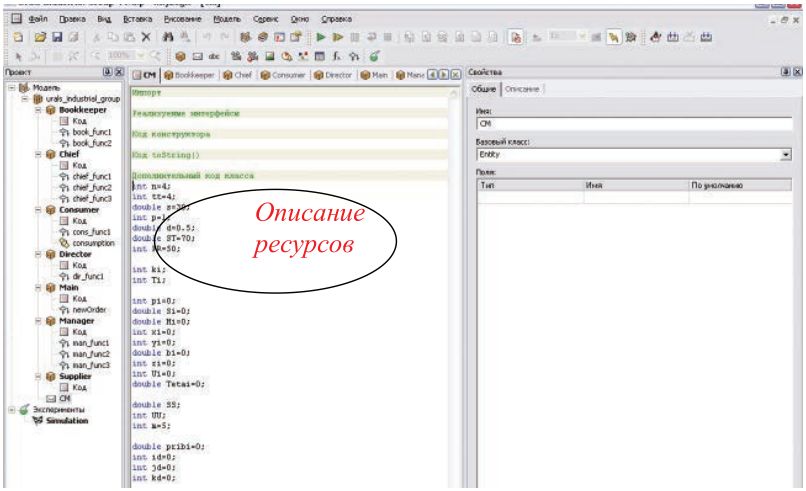


Рис. П.2.6. Описание ресурсов

• один тип / разные типы. Специальных средств описания типов ресурсов нет, можно создать любые типы ресурсов, описав объекты на языке Java.

Определение функции расчета параметра «штрафные санкции»

Можно описать расчет любых параметров функциями на языке Java. На реальном предприятии возможно отклонение фактического времени выполнения заказа от планового. Это может произойти по ряду причин, в первую возникают очередь задержки на исправление брака. За каждый день фактической задержки начисляется штраф, который затем выплачивается потребителю. Приведен пример реализации этого правила агентом – менеджером с помощью функции man_func3. Здесь *Hi* – размер штрафа по *i*-му заказу, *Si* – стоимость *i*-го заказа, *j*-е текущее модельное время, Tetai – время начала выполнения заказа, *Ti* – плановое время выполнения заказа.

Функция man_func3

```
if (mess.ri = = 2)
{
double j;
j = getTime();
if (j-mess.Tetai > mess.Ti)
{
mess.Hi = 0.001*mess.Si*(j-mess.Tetai-mess.Ti);
}
}
return 2;
```

Иерархическая структура модели мультиагентного процесса преобразования ресурсов поддерживается.

Элементы слияния и разветвления процессов. В данном пакете предусмотрено слияние и разветвление потоков ресурсов. Например, рассчитанное агентом–менеджером плановое время задержки выполнения заказа и его стоимость сообщаются потребителю, который принимает окончательное решение о размещении заказа на предприятии. В случае отказа потребителя информация поступает агенту – директору для анализа причин отказа. В случае согласия потребителя на условия размещения заказа (по стоимости и по времени выполнения) он перечисляет денежные средства на расчетный счет

предприятия (моделируется в виде сообщения бухгалтеру с задержкой на банковские дни). При этом заказ запускается в производство. Эти действия описывает функция man_func2. Здесь *Si* – стоимость *i*-го заказа, *SS* – предельная стоимость для потребителя, *Ui* – рассчитанное время задержки выполнения заказа (из-за нехватки ресурсов), *UU* – предельное время задержки потребителя.

Функция man_func2

```
if ((mess.Si < = mess.SS)&&(mess.Ui < = mess.UU))
{ return 1; }
else {return 3;}
```

Блок разветвления потоков ресурсов показан на рис. П.2.7.

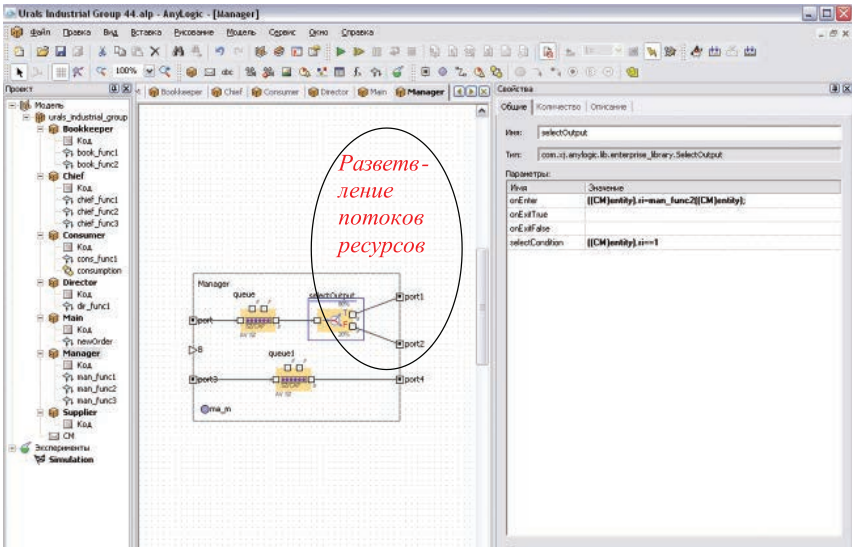


Рис. П.2.7. Разветвление потоков ресурсов

Функционально-стоимостный анализ. Система AnyLogic не имеет специальных средств поддержки функционально-стоимостного анализа.

Динамический анализ данных. В системе можно создавать различные графики и диаграммы, отражающие изменения их состояний в модельном времени (рис. П.2.8).

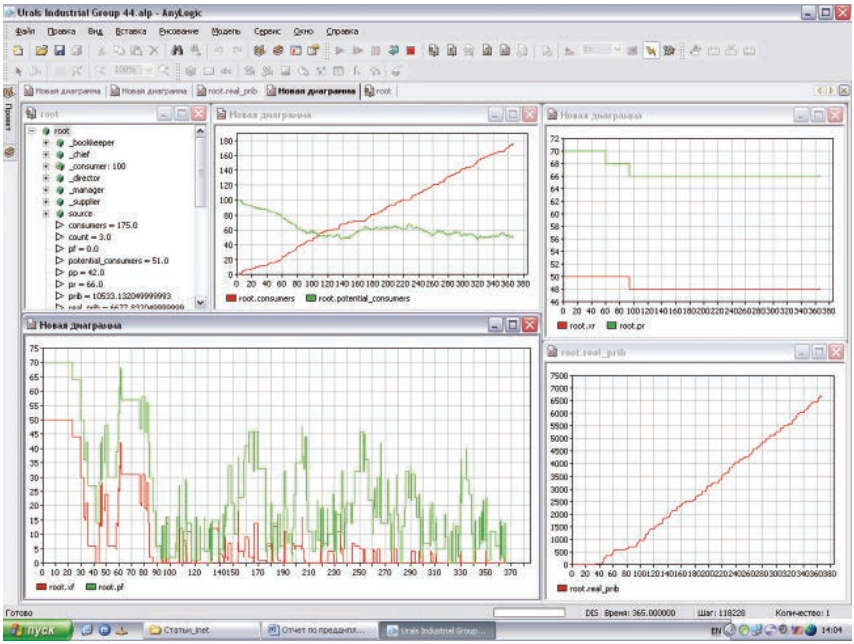


Рис. П.2.8. Средства динамического анализа данных

На рис. П.2.8 представлены диаграммы, отображающие динамику следующих переменных: количества потребителей и потенциальных потребителей; свободных производственных мощностей и основных рабочих; прибыли предприятия; количества производственных мощностей и основных рабочих.

Возможность построения мультиагентных моделей. На рис. П.2.1 представлены агенты модели базового предприятия, каждый из которых обладает индивидуальным поведением и знаниями. На основе описанных классов агентов можно создавать экземпляры класса с индивидуальным поведением. Сценарии поведения агентов и база знаний агентов описываются на языке высокого уровня (ЯВУ) Java, обмен сообщениями между агентами также описывается на языке Java.

Рассмотрим подробнее деятельность агента «chief – начальник производства». Он запускает заказы в производство, при этом задействуется необходимое количество производственных мощностей и трудовых ресурсов. При запуске заказа в производство агент

«начальник производства» осуществляет важную функцию планирования. Он планирует количество занятых производственных мощностей в массиве плановой загрузки по дням. Этот массив использует агент «менеджер» для расчета времени задержки выполнения заказа и согласования его с агентом «потребителем». Наличие планирования обуславливает интеллектуальное поведение агента.

Фрагмент поведения агента «начальник производства» описано функцией chief_func1. Вызов функции показан на рис. П.2.9.

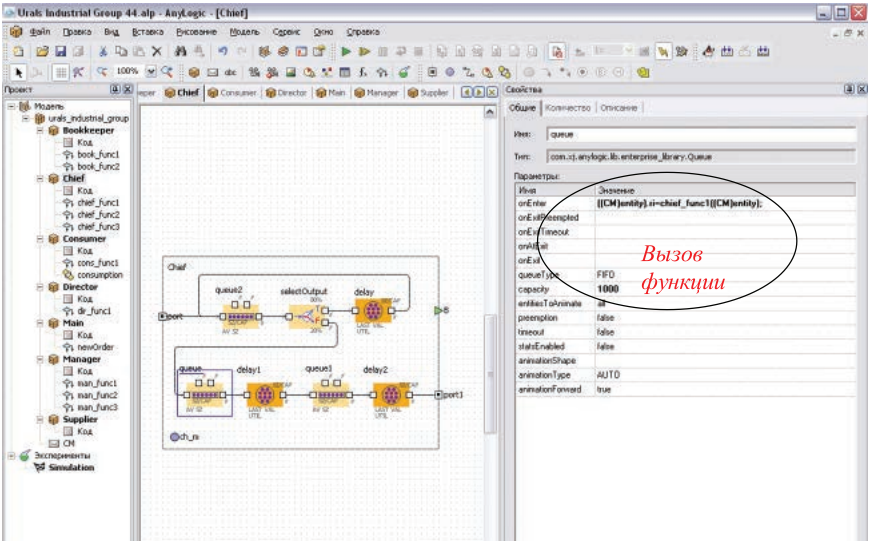


Рис. П.2.9. Структура Агента «chif – начальник производства»

Функция chief_func1

```
if (mess.pi < = ch_m.pf)
{
ch_m.pf = ch_m.pf-mess.pi;
ch_m.xf = ch_m.xf-mess.xi;
if (ch_m.xf < 0)
{
mess.yi = -ch_m.xf;
mess.xi = mess.xi + ch_m.xf;
ch_m.xf = 0;
```

```
}
mess.Tetai = getTime();
double k = getTime();
int j = (int)k;
int i = 1;
 while (i < = mess.Ti)
{
double r = get(B,(j-1)) + mess.pi;
 B.set((j-1),r);
i + + ;
j + + ;
}
}
return 2;
```

Заполнение массива планирования ресурсов «В»

Поддержка русского языка. Русский языкподдерживается минимально. При описании агентов, классов, переменных, параметров и т. д. русский язык не поддерживается.

Изменение параметров модели во время эксперимента. При проведении эксперимента есть возможности остановить имитацию и изменить характеристики и/или параметры.

Удобство интерфейса пакета при описании элементов процесса преобразования – дружественный интерфейс, ориентированный на программиста.

Поддержка аппарата ЭС. Система AnyLogic не имеет специальных средств описания базы знаний о предметной области и механизма вывода. Описание моделейагентов выполнено наJava.

Проектирование концептуальной модели предметной области не поддерживается.

Описание целей системы. Система AnyLogic не имеет специальных средств описания целей системы или отдельных ее элементов.

Наличие механизмов описания ситуаций (поддержка ситуационного подхода). AnyLogic не имеет специальных средств описания ситуаций.

Наличие средств описания команд. Команды описываются на языке Java.

Описание модели на ограниченном естественном языке. Отсутствуют.

Поддержка ИМ – система AnyLogic является СИМ.

Поддержка аппарата ЭС. Система AnyLogic не имеет специальных средств интеграции с ЭС и аппарат ЭС не поддерживает.

Удобство интерфейса пакета при описании агентов – дружественный интерфейс, ориентированный на программиста.

Экспорт / импорт / открытый доступ к базе знаний модели:

- *импорт начальных условий и исходных данных из внешних источников* – существует механизм настройки экспорта из внешних баз данных;
- *экспорт результатов экспериментов* – механизм отсутствует, но при необходимости его можно описать функциями на языке Java;
- *открытый доступ к базе знаний модели* – отсутствует.

Приложение 3

Экспертная система реального времени G2

Модель в G2 описывается в виде блоков, представляющих собой какие-либо задачи (работы), связанные между собой путями («дорожками»), определяющими последовательность выполнения операций. На рис. П.3.1 представлены основные процессы предприятия.

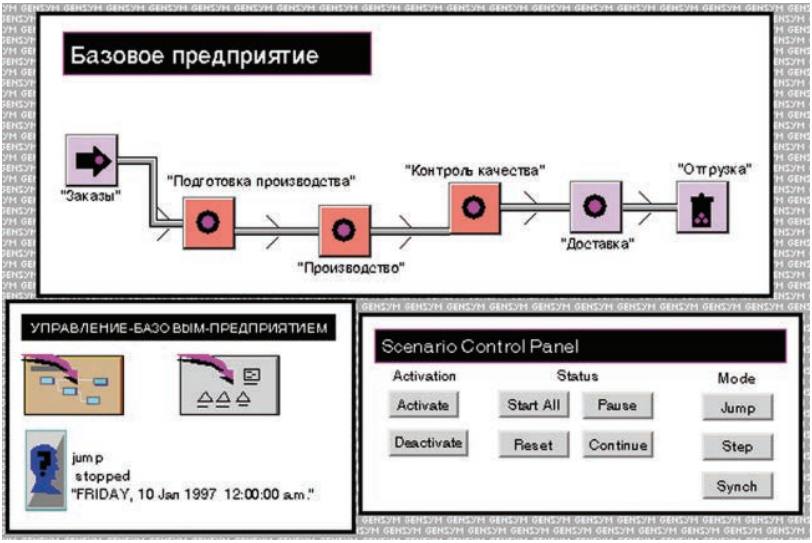


Рис. П.3.1. Модель абстрактного предприятия на G2

В модели на базе G2 существует понятие рабочего объекта. Данный вид объектов модели является динамическим и представляет собой такой объект, обработка которого и происходит в процессе моделирования (заказы, расходные материалы, готовые продукты). Рабочие объекты передаются в модели от объекта к объекту по установленным путям и подвергаются некоторым изменениям / преобразованиям.

Модель преобразователя «выполнение заказов на производстве» (рис. П.3.2).

- вектора входов / выходов / средств представлены входами («заказы на выполнение»; выходами («выполненные заказы»); средствами («Мощности», «Рабочие»).

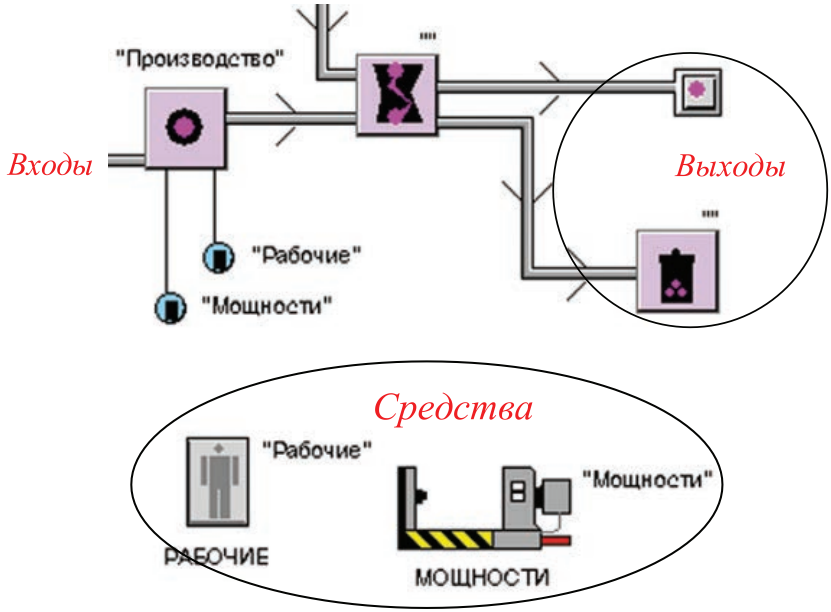


Рис. П.3.2. Вектора входов/выходов/средств

Для выполнения тех или иных операций может потребоваться описание необходимых ресурсов / средств (рабочие, станки, транспорт, ЭВМ и т. п.). В G2 данный вид объектов представлен набором подклассов класса ресурсов (рис. П.3.3).

● объем захватываемого / освобождаемого ресурса / средства описывается количеством средств на обработку одного объекта (партии заказа). Кроме того, существует возможность задавать правило выбора средств из числа свободных на данный момент (первый попавшийся, по приоритету, по наименьшей стоимости пользования или наименее редко используемого). Захват / освобождение необходимого количество ресурсов реализуется внутренними механизмами G2.

● длительность операции задается в виде постоянной величины или величины с заданным отклонением. Каждый заказ перед производством разбивается на партии. На рис. П.3.4 длительность равна значению переменной *t*, т. е. задается норматив производства 1 партии;

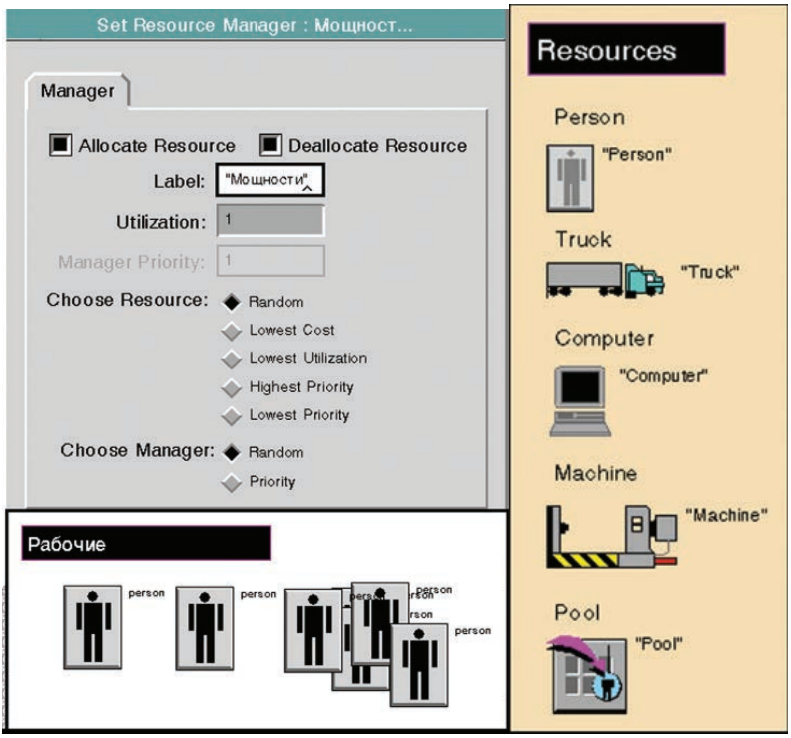


Рис. П.3.3. Настройка используемых ресурсов/средств

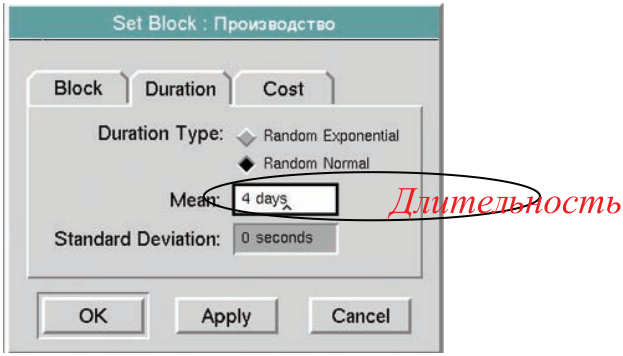


Рис. П.3.4. Длительность операции

• приоритет операции не устанавливается, обрабатывается очередь поступивших на производство партий (первым пришел – первым ушел);

• условие запуска преобразователя (рис. П.3.5) реализуется внутренними механизмами G2. Таким образом, выполнение работы инициируется поступлением на вход блока рабочего объекта.

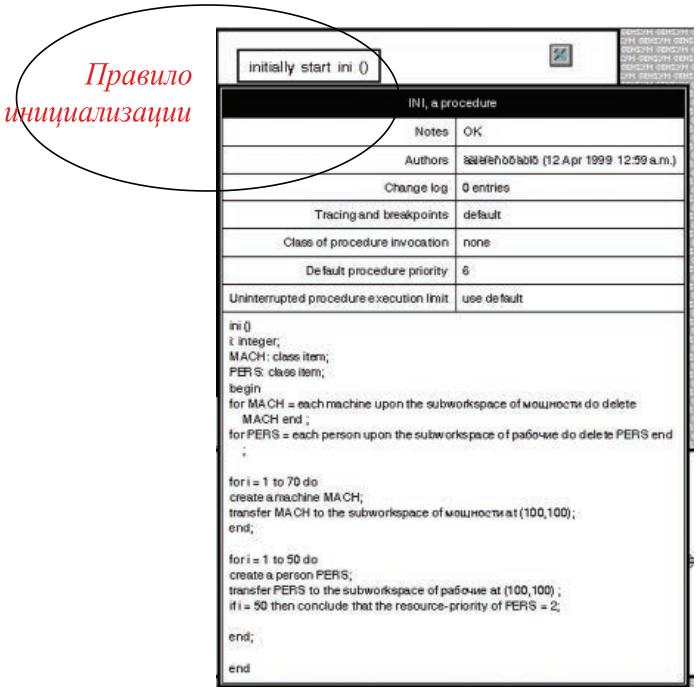


Рис. П.3.5. Инициализация ресурсов

Определение функции расчета параметра «штрафные санкции».

Расчет любых параметров описывается функциями, а при наличии условий при расчете необходимо воспользоваться механизмом процедур, инициируемым правилом. Правила в G2 позволяют отслеживать некоторые события модели и инициировать операции.

Иерархическая структура модели процесса преобразования ресурсов. Поддерживается.

Элементы слияния и разветвления процессов. В G2 предусмотрено слияние и разветвление потоков ресурсов. Для реализации разветвления используется 2 типа блоков:

• блок «Copy» осуществляет безусловное копирование входящего рабочего объекта на каждый из своих выходов. Для синхронизации разветвленных потоков рабочих объектов далее в модели используются блоки «Associate» и «Reconcile». «Associate» связывает объекты, а «Reconcile», получая на вход 2 потока объектов, далее пропускает парами и только связанные объекты. При этом каждый входящий рабочий объект дожидается свою пару в очереди блока;

• блок «Branch» – имитирует процесс принятия решения и перемещает входящий рабочий объект на один из своих выходов, в зависимости от условия. В качестве условия возможен анализ атрибутов рабочего объекта.

Блоки разветвления потоков ресурсов показаны на рис. П.3.6.

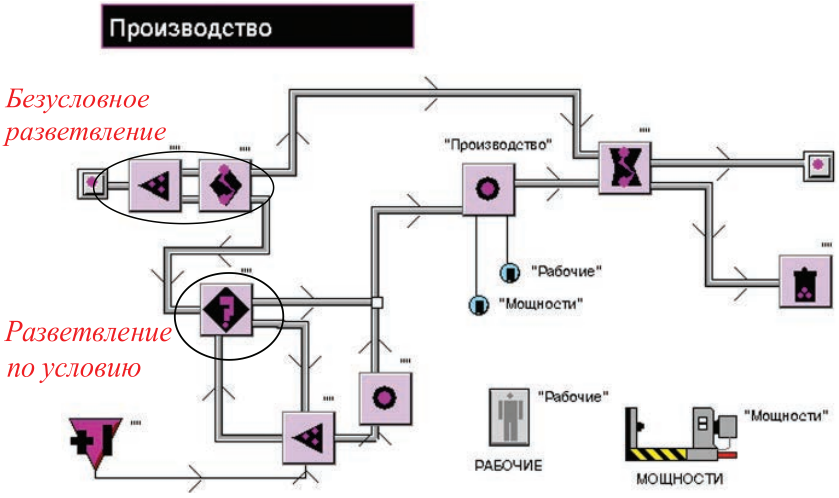


Рис. П.3.6. Разветвление потоков ресурсов

Синхронизация по времени процессов. G2 имеет специальные средства поддержки синхронизации по времени процессов (рис. П. 3.1). Реализованы режим синхронизации с реальным временем и пошаговый режим, который, в свою очередь, делится на временные шаги и на шаги по событиям.

Функционально-стоимостный анализ. G2 имеет специальные средства поддержки функционально–стоимостного анализа. Каждый блок, ресурс и рабочий объект модели имеет встроенные средства учета стоимости

работ, совершенных блокомнад рабочим объектом (рис. П.3.7). При построении модели задается стоимость обработки рабочего объекта в каждом блоке, а при моделировании ведется автоматический учет стоимости.

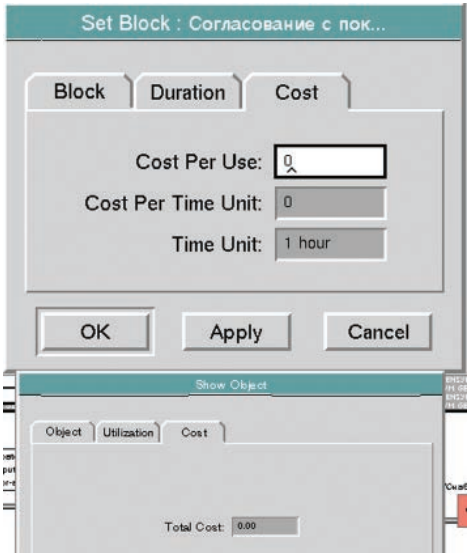


Рис. П.3.7. Элементы функционально-стоимостного анализа

Время работы / прерывания / простоя. Длительность операции задается в виде постоянной величины или величины с заданным отклонением (рис. П.3.4).

Динамический анализ данных. Проанализируем возможности средств анализа динамических данных.

На рис. П.3.8 отражены варианты представления результатов моделирования. В G2 существует класс датчиков, которые могут быть подключены к любому статичному объекту (блок, средства и т. п.), и настроены на накопление истории изменения любого из атрибутов динамических рабочих объектов модели. История значений датчика может быть представлена в табличном виде или в виде графика значений. Кроме того, представление информации может быть организовано как в виде индивидуальных таблиц/графиков по каждому атрибуту, так и в виде общего, где значения всех или части датчиков отражается в одной таблице / графике.

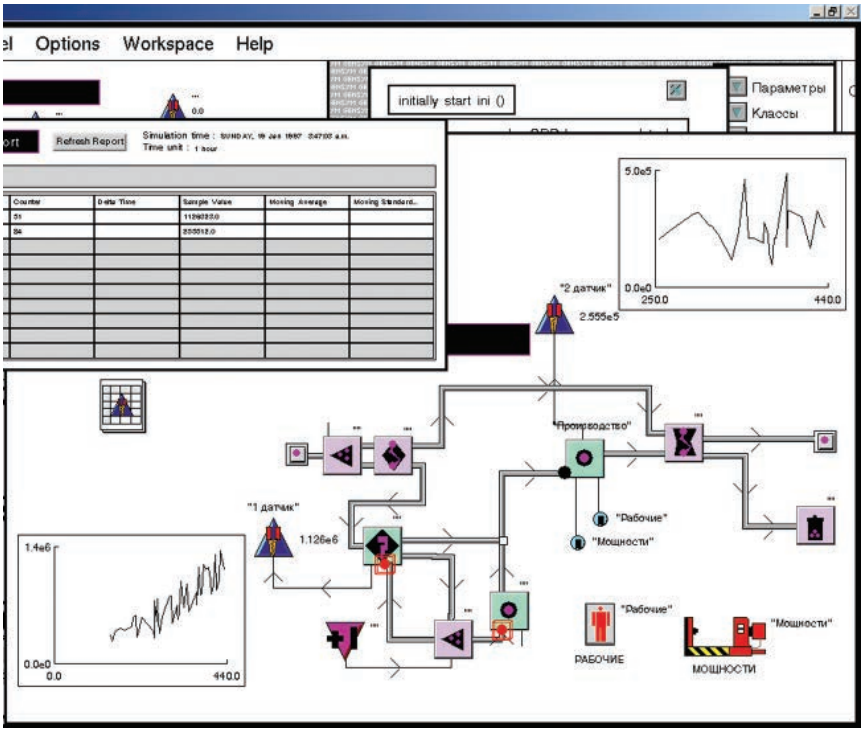


Рис. П.3.8. Отображение результатов моделирования

В процессе моделирования вся динамика движения заказов и готовой продукции наглядно реализована средствами среды G2.

Поддержка русского языка. Русский язык поддерживается только для обозначения элементов модели (наименования объектов и их атрибутов).

Изменение параметров модели во время эксперимента. При проведении эксперимента в любое время можно остановить имитацию и изменить характеристики и/или параметры.

Возможность построения мультиагентных моделей. Существует возможность разработать описания агентов, каждый из которых обладает индивидуальным поведением и знаниями. На основе описанных классов агентов можно создавать экземпляры класса с индивидуальным поведением (рис. П.3.9).

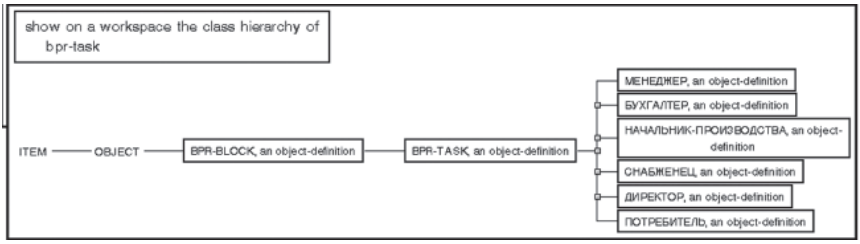


Рис. П.3.9. Отображение иерархии классов агентов

Поведение экземпляров задается структурой агента и набором правил, касающихся именно этого класса агентов. Под структурой подразумевается возможность детализации агента, путем разделения операций агента на составляющие (рис. П.3.10).

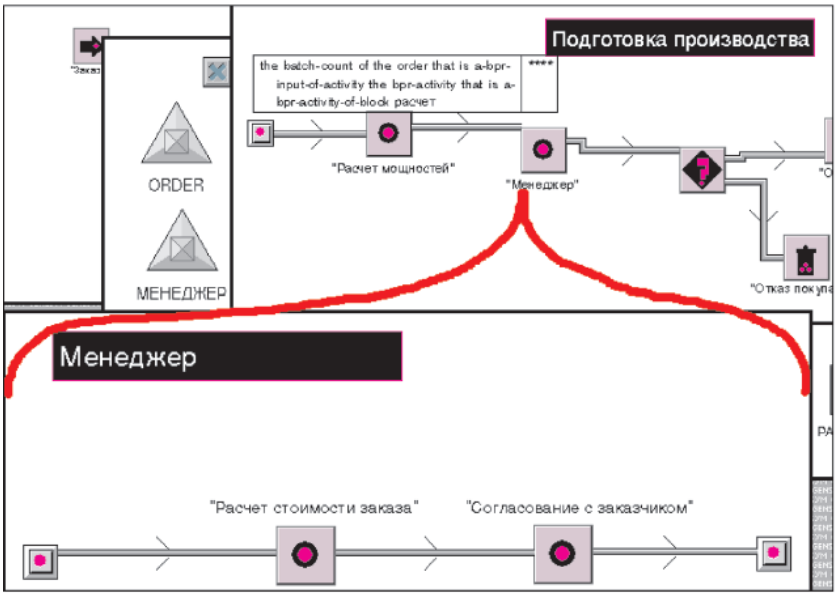


Рис. П.3.10. Отображение структуры агента

База знаний агентов описывается правилами G2 (рис. П.3.11). Правила G2 – это функции поведения объектов модели, описанные

на ограниченном естественном языке (английский), которые определяют, как модель будет реагировать на факт выполнения каких-либо условий.

Обмен сообщениями между агентами может быть реализован несколькими способами как через состояние некоторых объектов модели, так и через содержание переменных / массивов модельной среды.

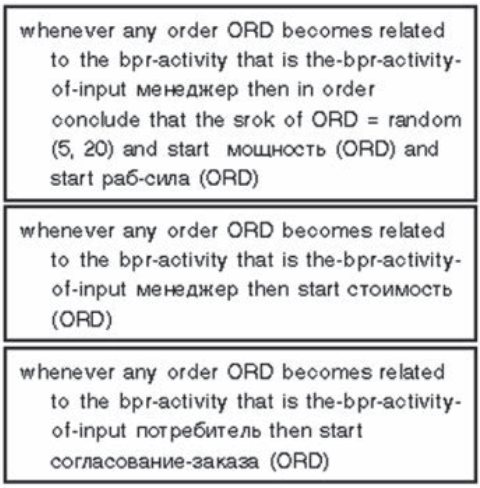


Рис. П. 3.11. Фрагмент базы знаний
(3 правила поведения агентов менеджер и потребитель)

Описание базы знаний о предметной области. G2 имеет специальные средства описания базы знаний. Знания описываются в виде правил на языке высокого уровня, приближенного к английскому (рис. П.3.12).

Описание целей системы. G2 не имеет специальных средств описания целей системы или отдельных ее элементов. При необходимости цели можно описать функциями на языке G2.

Наличие механизмов описания ситуаций (поддержка ситуационного подхода). G2 не имеет специальных средств описания ситуаций. При необходимости их можно описать на языке G2.

Наличие средств описания команд. Команды описываются на собственном языке G2.

Описания модели на ограниченном естественном языке. Для описания модели используется приближенный к естественному английский язык.

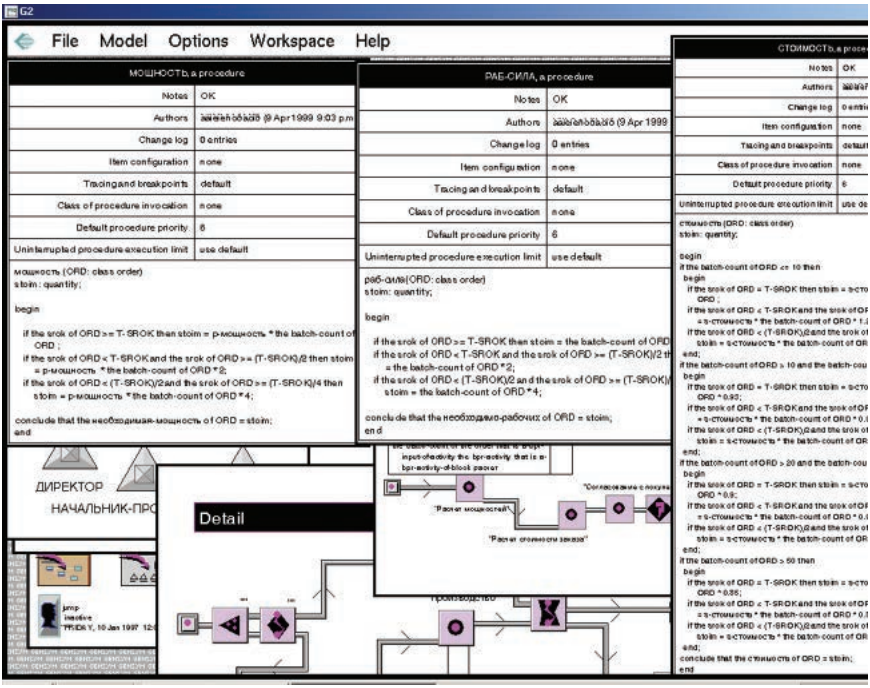


Рис. П. 3.12. Функции поведения агентов, вызываемые правилами

Поддержка аппарата ЭС – да.

Поддержка ИМ. Реализовано.

Удобство интерфейса пакета при описании агентов – дружественный английский интерфейс.

Экспорт/ импорт/ открытый доступ к базе знаний модели:

● импорт начальных условий и исходных данных из внешних источников – существует механизм настройки экспорта из внешних баз данных, текстовых файлов и т. п.;

● экспорт результатов экспериментов – существует модуль двухстороннего взаимодействия с базами данных и встроенные механизмы работы с текстовыми файлами;

● открытый доступ к базе знаний модели – отсутствует.

Проектирование концептуальной модели предметной области – не поддерживается.

Приложение 4

Система моделирования и проектирования ARISToolSet

На рис. П.4.1 представлена модель абстрактного предприятия в ARIS.

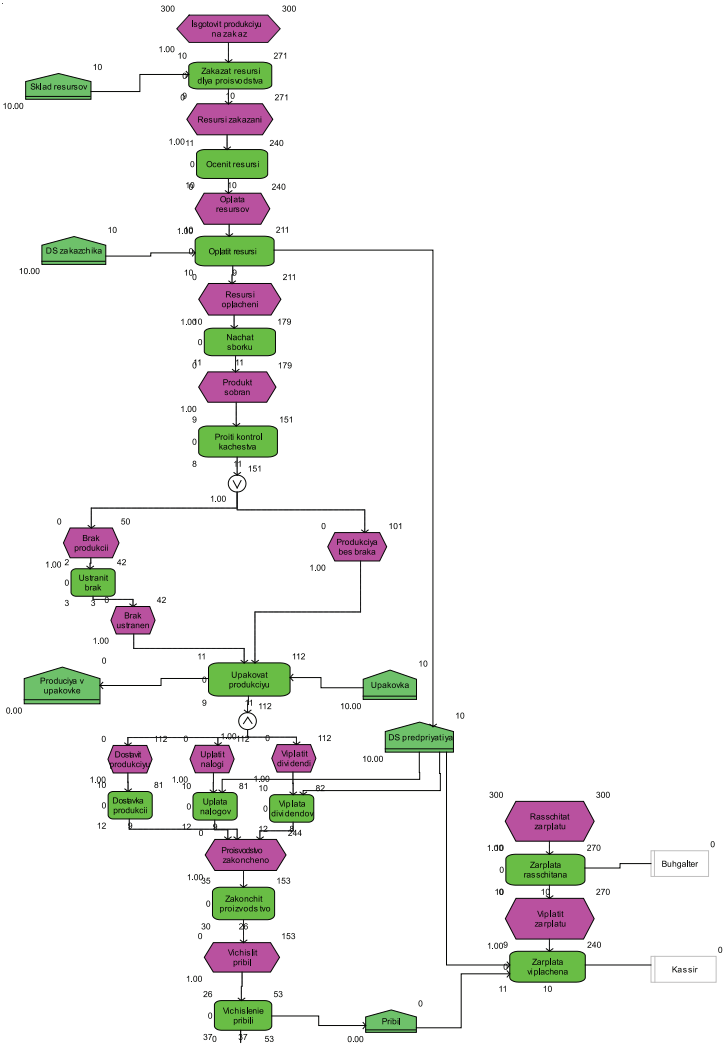


Рис. П. 4.1. Модель абстрактного предприятия в ARIS

Модель преобразователя «выплата зарплаты» представлена на рис. П.4.2.

● вектор входов/выходов/средств представляется отдельными связями с соответствующими ресурсами и средствами;

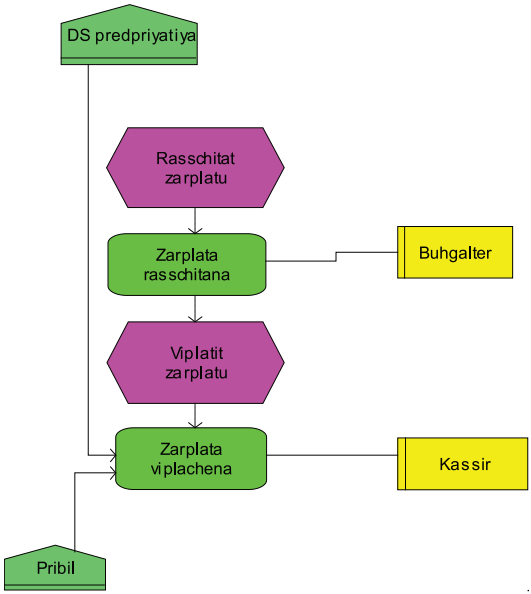


Рис. П.4.2. Операция «Выплата зарплаты»

● объем ресурсов и средств – величина постоянная. Случайная величина и функция в ППП ARIS не предусмотрена. Так, при расчете зарплаты используется один бухгалтер (рис. П.4.3);

	Buhgalter [English]
Number of employees	1
Priority	

Рис. П.4.3. Количество бухгалтеров

● длительность операции: const/функция / случайная величина (рис. П.4.4);

	Poluchit DS na resursi [English]
Static wait time	Constant(0000:00:10:00)
Orientation time	Normal distribution(mu = 0000:00:10:00 , sigma = 0000:00:00:30)
Processing time	Constant(0000:00:10:00)
Orientation (always)	☑ Orientation (always)
Interruptable	☑ Interruptable
Processes to be process	1
Resource allocation	Additive

Рис. П.4.4. Описание длительности операции

• интенсивность преобразования в ППП ARIS не предусмотрена;

• тип приоритета поддерживается только относительный (рис. П.4.5);

	Prichla zayavka na proizvodstvo [English]
Probability	
Priority	1
Comparison operator	
Comparison value	
Comparison value (num	1.00
Comparison value (logic	☑ Comparison value (logical)
Commit resource	☑ Commit resource

Рис. П.4.5. Приоритет операции

• условие запуска преобразователя определяется как функция времени (рис. П.4.4) и наличия необходимого количества входных ресурсов и средств.

Определение функции расчета характеристики – не предусмотрено.

Элементы слияния и разветвления процессов – могут быть использованы логические операторы, такие как «и», «или», «исключающее ИЛИ» (рис. П.4.6).

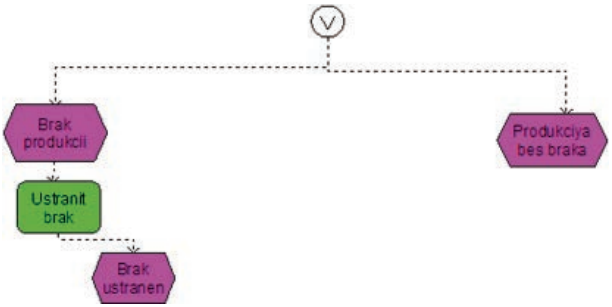


Рис. П. 4.6. Слияние и разветвление

Динамический анализ данных. Пример графика представлен на рис. П.4.7.

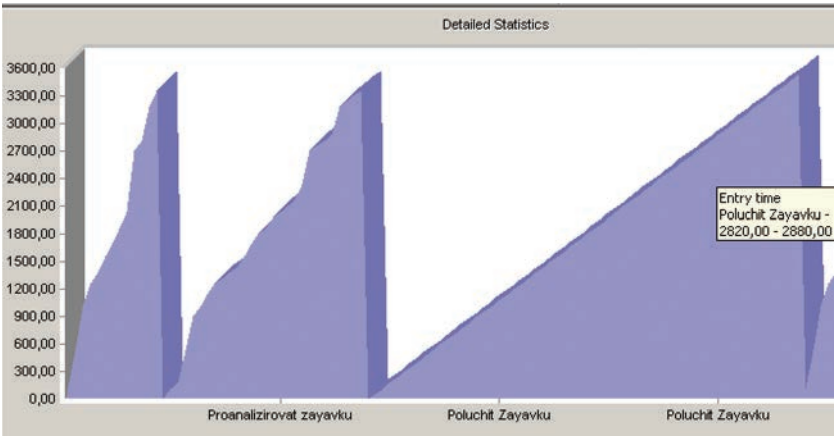


Рис. П.4.7. Построение диаграмм и графиков

Синхронизация по времени – может быть использован блок «И», формирующий сообщения.

Функционально-стоимостный анализ. Стоимостные характеристики задаются в настройках операции (вкладка Costs).

Иерархическая структура процессов. Количество уровней в ARIS – не ограничивается, на рис. П.4.2 представлен пример декомпозиции процесса выплаты зарплаты. Расчет параметров процесса отсутствует.

Поддержка русского языка. Русский язык не поддерживается.

Методика *BSC* – поддерживается в модуле ARIS Balanced Score Card (рис. П.4.8), но не интегрирована с модулем ARIS Simulate.

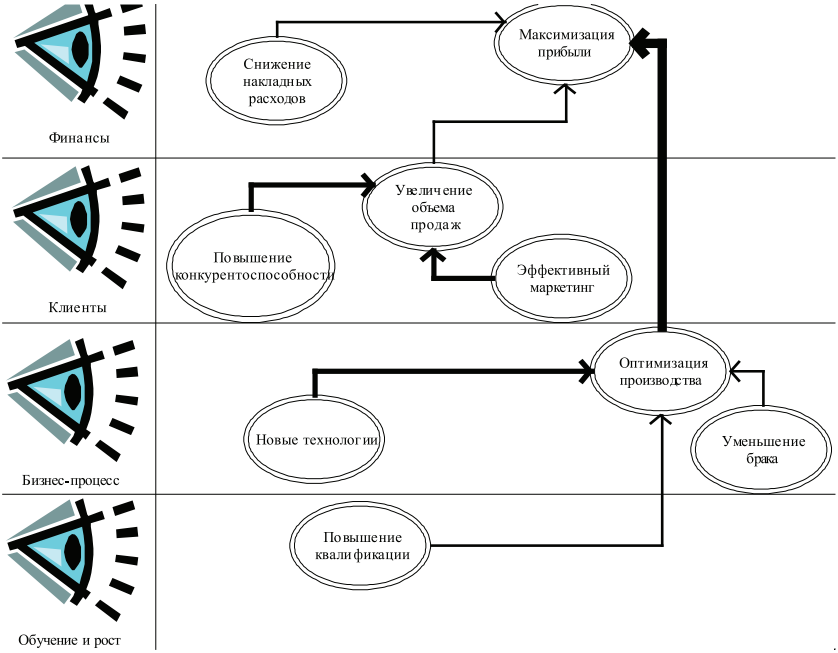


Рис. П.4.8. Граф причинно-следственных связей BSC

Изменение параметров модели во время экспериментов. ППП ARIS не позволяет изменять настройки модели во время эксперимента.

Проектирование концептуальной модели предметной области – не поддерживается.

Возможность создания мультиагентных моделей – отсутствует.

Поддержка ИМ – реализовано в модуле ARIS Simulate.

Поддержка аппарата ЭС – отсутствует.

Приложение 5

СДМС BPsim2

На рис. 4.2 представлена модель базового предприятия. В модели базового предприятия описаны агенты: потребитель, менеджер, бухгалтер, начальник производства, управляющий поставками (снабженец), директор. Агент директор представлен на рис. П.5.1.

Рис. П.5.1. Верхний уровень модели базового предприятия

Модель процесса производства представлена на рис. П.5.2.

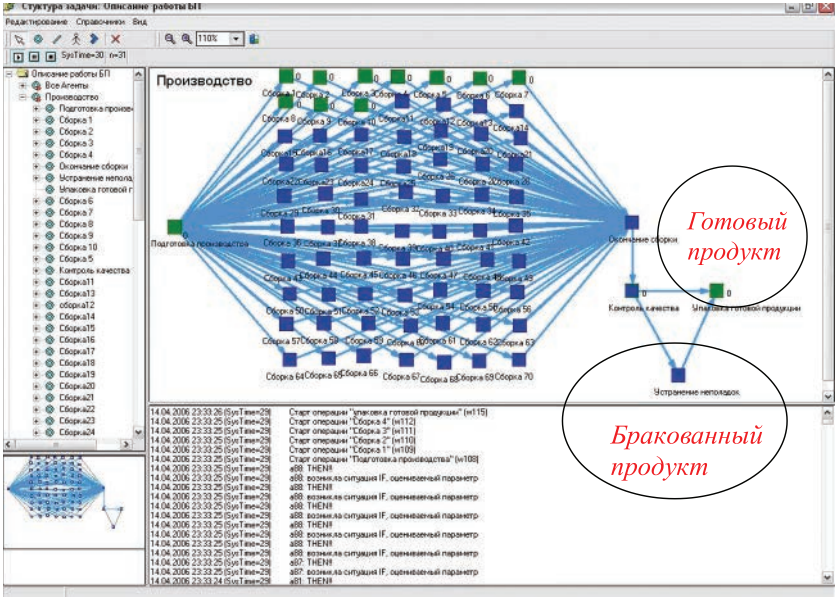


Рис. П.5.2. Выполнение заказов на производстве

• вектора входов / выходов / средств представлены входами («заказы на выполнение»); выходами («выполненные заказы»); средствами («количество свободных производственных мощностей», «количество свободных рабочих»);

• объем захватываемого / освобождаемого ресурса / средства может быть постоянная / функция / случайная величина. Необходимое количество ресурсов описывается в виде функции или точного количества необходимых ресурсов, как показано на рис. П.5.3.

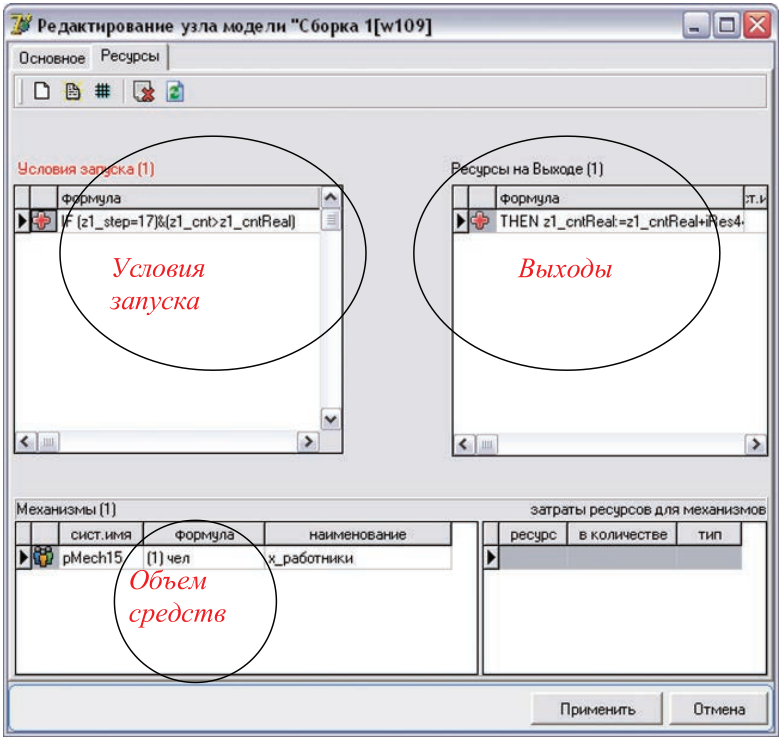


Рис. П.5.3. Вектора входов/выходов/средств

В качестве средств могут быть описаны: исполнители работы, оборудование, аппаратное обеспечение, программное обеспечение, машинный ресурс. С помощью формулы можно ввести необходимое количество ресурсов. Справочник ресурсов представлен на рис. П.5.4.

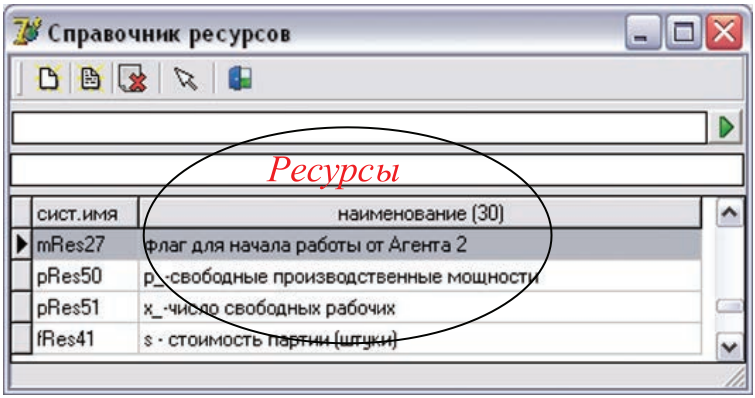


Рис. П.5.4. Справочник ресурсов

• длительность операции: постоянная / функция / случайная величина. Длительность операции может быть определена в виде функции, константы и случайной величины. Например, на рис. П.5.5 длительность равна 2;

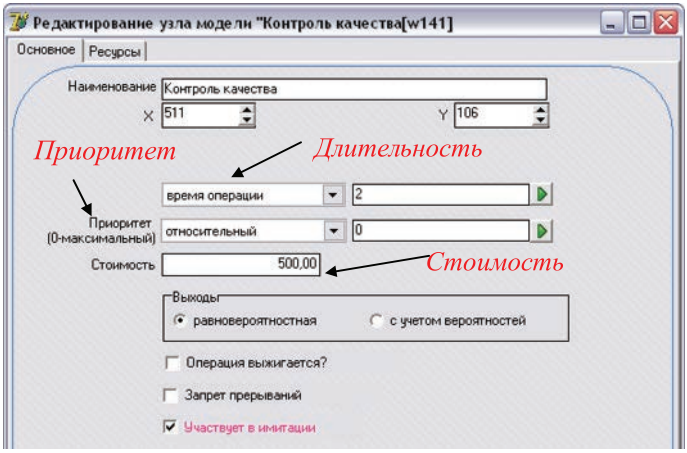


Рис. П.5.5. Длительность и приоритет операции

• приоритет операции и тип приоритета описываются на вкладке «Основное» формы редактирования узла модели (см. рис. П.5.5). Тип приоритета может быть как относительным, так и абсолютным.

Система поддерживает остановку выполнения операции (поступление на вход операции специального сообщения). Приоритет необходим для разрешения конфликтов, которые возникают на общих ресурсах и средствах. Абсолютный приоритет подразумевает прерывание выполнения запущенных элементов с целью освобождения средств, необходимых для выполнения элемента с большим приоритетом;

- условие запуска преобразователя в целом рассматривается как функция времени, ресурсов и средств. Так, на рис. П.5.3. представлено описание условия запуска процесса сборки.

Модель ресурса и средства:

- описывается вектором;
- существуют ограничения: минимум, максимум;

Ресурсы используются при описании функций (рис. П.5.6). При этом доступен справочник значений ресурсов. Так, например, перед началом моделирования pRes51 «количество свободных рабочих» равно 50;

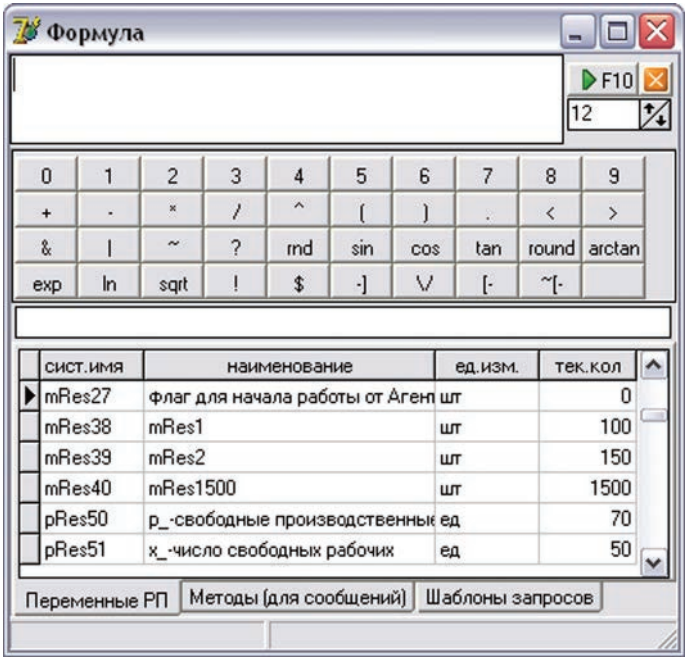


Рис. П.5.6. Механизм описания моделей ресурсов

• один тип / разные типы (рис. П.5.7), как видно из рисунка, поддерживаются следующие типы ресурсов: информационные, материальные, финансовые, трудовые:

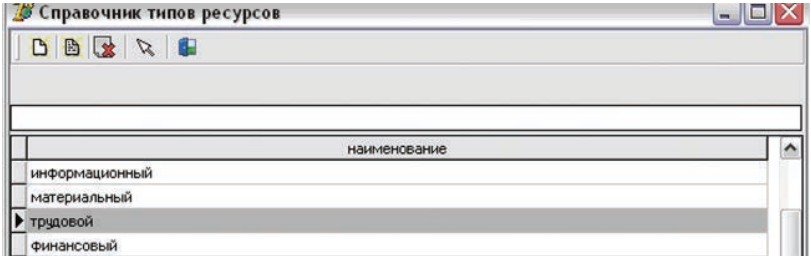


Рис. П.5.7. Справочник типов ресурсов

Определение функции расчета параметра «стоимость»

На рис. П.5.5 приведен пример описания параметра «стоимость». Стоимость заказа рассчитывается многовариантно, в зависимости от требований Потребителя. На рис. П.5.8 представлено описание расчета стоимости заказа при конкретных условиях. Таким образом, ППП BPsim2 позволяет определять функцию расчета данного параметра.

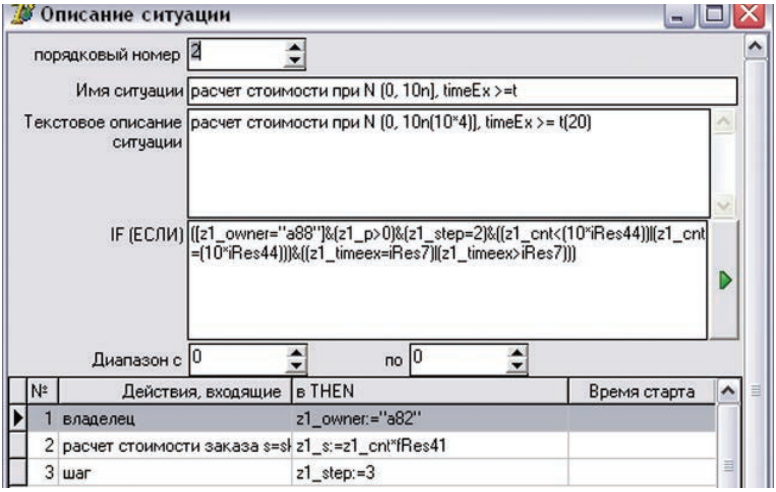


Рис. П.5.8. Описание расчета стоимости в конкретной ситуации

Иерархическая структура модели мультиагентного процесса преобразования ресурсов – поддерживается.

Элементы слияния и разветвления процессов. В данном пакете предусмотрено слияние и разветвление потоков ресурсов с помощью перекрестков. На рис. П.5.2 показана декомпозиция процесса «производство», организуется разветвление процесса и формируется два альтернативных потока: «бракованный продукт», и «готовый продукт».

Синхронизация по времени процессов. Система BpSim2 поддерживает синхронизацию по времени процессов (позволяет запускать одновременно несколько процессов), реализовано с помощью блоков синхронизации, формирующих одновременно несколько сигналов.

Функционально-стоимостный анализ. На рис. П.5.2 приведен пример описания стоимости операции «Контроль качества». Стоимость данной операции равна 500 рублей.

Проектирование концептуальной модели предметной области реализовано в оболочке ЭС «Конструктор фрейм-систем».

Динамический анализ данных. Возможности средств анализа динамических данных представлены на рис. 3.30.

Поддержка русского языка. Русский язык, как видно на всех рисунках, поддерживается корректно.

Изменение параметров модели во время эксперимента. При проведении эксперимента в любое время можно остановить имитацию, нажав кнопку «Пауза», и изменить характеристики и/или параметры.

Удобство интерфейса пакета при описании элементов ППР – дружественный интерфейс.

Возможность построения мультиагентных моделей поддержана. На рис. 4.1.2 и на рис. П.5.1 представлены все агенты модели базового предприятия (потребитель, менеджер, начальник производства, бухгалтер, управляющий поставками, снабженец, директор), каждый из которых обладает индивидуальным поведением и знаниями. На рис. П.5.9 представлено окно описания свойств агента – директора (показан список его целей).

Поддержка аппарата ЭС. На рис. П.5.10 представлено описание базы знаний агента – менеджера.

Описание целей системы. Система BPsim2 имеет в своем составе специальные средства описания целей в виде графов (см. рис. 3.23) и системы сбалансированных показателей (см. рис. 3.25).

На рис. П.5.11 представлены цели агента-директора в модели базового предприятия (в виде графа целей).

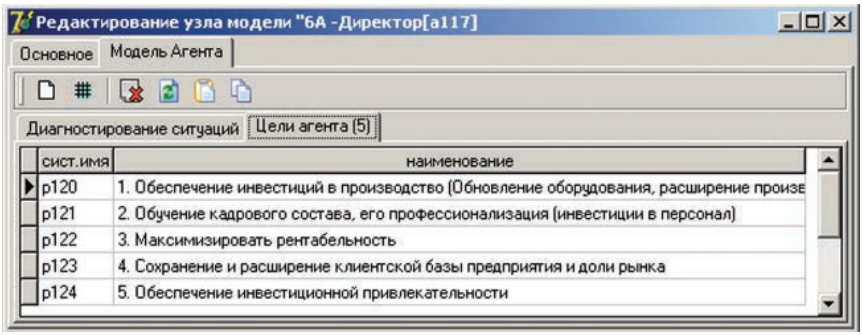


Рис. П.5.9. Свойства агента – менеджера

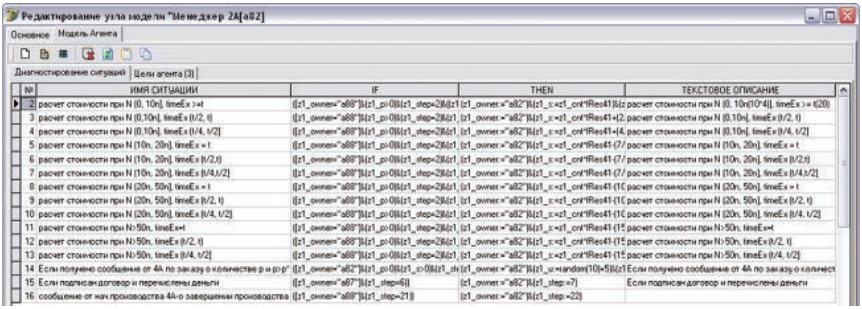


Рис. П.5.10. База знаний агента – менеджера

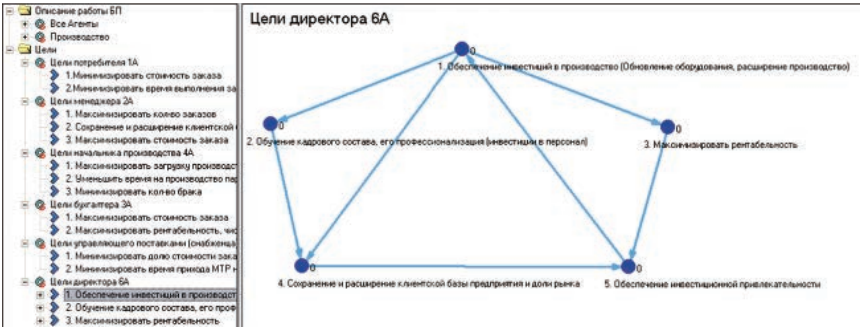


Рис. П.5.11. Описание графа целей «агента – директора»

На основе показателей и их нормативных значений, а также допустимых границ отклонений от целевого значения формируется «карта показателя», представленная на рис. 3.22. «Карта показателя» представляет собой шкалу для определения «степени достижения цели» агентом. В зависимости от изменений значений параметров во время моделирования будет меняться угол наклона стрелки, указывающей на одну из «зон». Красной зоне соответствует значение показателя, определяющее «далекое от достижения цели» состояние. Зеленой зоне соответствует «стабильное» состояние, в котором «цель достигается». Синей зоне соответствует «крайне высокая степень достижения цели».

Каждая цель связана с определенной областью (стратой). В среде BPsim возможно построение BSC. Она показывает взаимосвязь целей агента, одновременно отражая положение целей внутри всех страт. На рис. П.5.12 представлена BSC целей «агента – директора».

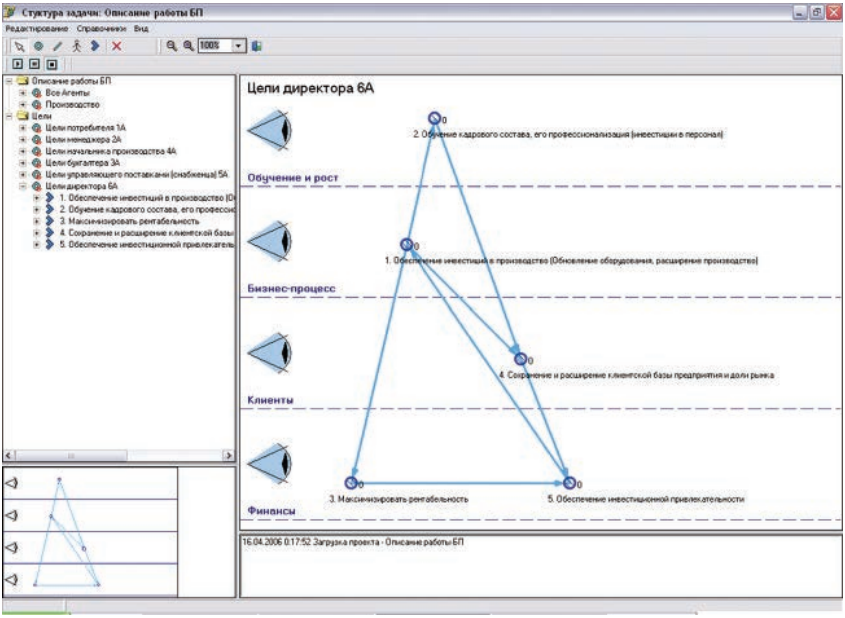


Рис. П.5.12. BSC целей «агента – директора»

Наличие механизмов описания ситуаций (поддержка ситуационного подхода). На рис. П.5.11 представлен пример описания ситуаций.

Наличие средств описания команд. Система BpSim2 имеет в своем составе специальные средства описания команд, представленные на рис. П.5.8, П.5.10.

Возможность описания модели на ограниченном естественном языке. ППП BPsim2 использует механизмы описания модели на ограниченном естественном языке, применяя интерфейсы, ориентированные на конечного пользователя, и логику предикатов первого порядка.

Поддержка ИМ в системе BPsim2 реализована, так как система построена на основе СИМ BPsim.

Удобство интерфейса пакета при описании агентов – дружественный интерфейс.

Экспорт /импорт/ открытый доступ к базе знаний модели:

– импорт начальных условий и исходных данных из внешних источников – существует встроенный механизм настройки импорта из внешних баз данных, реализованный на основе поддержки языка Transact-SQL, пример представлен на рис. П.5.13;

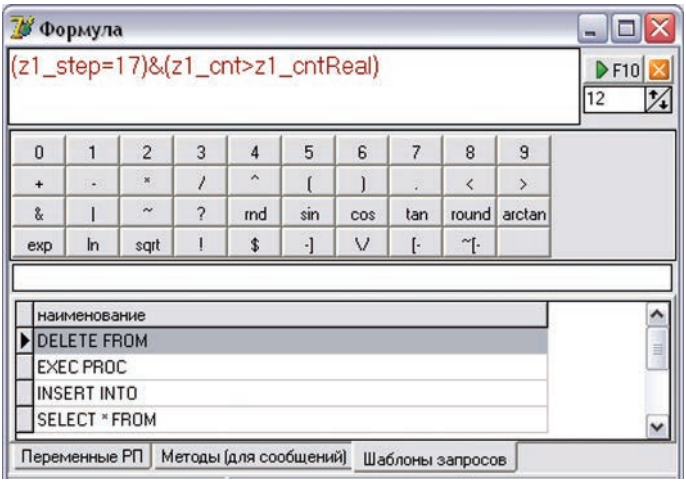


Рис. П.5.13. Шаблоны запросов модели базового предприятия

– экспорт результатов экспериментов – существует механизм экспорта результатов экспериментов во внешние средства анализа данных (MS Excel и MS Project);

– открытый доступ к базе знаний модели – модель существует в виде базы данных MS SQL Server и может быть доступна ее средствами.

Приложение 6

Результаты сравнения систем

Результаты сравнения функциональных возможностей пакетов представлены в табл. П.6.

Таблица П.6

Сравнительный анализ систем, близких по функциональности к СДМС

№ п/п	Параметр	ARIS	G2	AnyLogic	BPsim	BPsim2
1	Проектирование концептуальной модели предметной области	НЕТ	НЕТ	НЕТ	НЕТ	+
2	Язык описания процессов преобразования ресурсов					
2.1	– Описание ресурсов, средств, преобразователей	+	+	+	+	+
2.2	– Описание целей системы – в виде графа; – в виде BSC.	 + +	 + НЕТ	 НЕТ НЕТ	 НЕТ НЕТ	 + +
2.3	– Иерархическая модель процесса	+	+	+	+	+
3	Наличие языка описания команд	НЕТ	+	НЕТ	НЕТ	+
4	Описание модели на ограниченном естественном языке	НЕТ	+	НЕТ	+	+
5	Построение мультиагентной модели					
5.1	– элемент АГЕНТ	НЕТ	НЕТ	+	НЕТ	+
5.2	– модели поведения агентов	НЕТ	НЕТ	+	НЕТ	+
5.3	– база знаний агента	НЕТ	НЕТ	НЕТ	НЕТ	+
5.4	– язык обмена сообщениями	НЕТ	НЕТ	НЕТ	НЕТ	+
6	Имитационное моделирование	+	+	+	+	+
7	Экспертное моделирование	НЕТ	+	НЕТ	НЕТ	+
8	Ситуационный подход	НЕТ	+	НЕТ	НЕТ	+

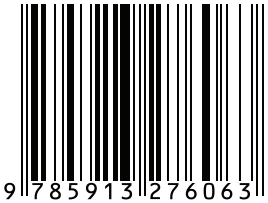
Как следует из табл. П.6 и проведенного сравнительного анализа, система BPsim2 обладает полной функциональностью мультиагентной СДМС процессов преобразования ресурсов. На основе систем G2 и AnyLogic возможно построение мультиагентных СДМС, причем значительно меньшие усилия потребуются при использовании G2, так как данная система поддерживает аппарат ЭС. Проектирование концептуальной модели предметной области и построение мультиагентных моделей, содержащих интеллектуальных агентов, поддерживает только система BPsim2. Специализированными средствами поддержки методики BSC обладают только системы ARIS и BPsim2, но ARIS не поддерживает интеграции имитационного моделирования и BSC. Понятийный аппарат всех рассмотренных систем соответствует проблемной области процессов преобразования ресурсов. Описание модели на ограниченном естественном языке поддерживается в системах G2 и BPsim2. С точки зрения пользователя, не обладающего навыками программирования, удобными средствами описания / создания модели мультиагентного процесса преобразования обладает только проблемно-ориентированная СДМС BPsim2. В системах AnyLogic и G2 при создании сложных мультиагентных моделей графических средств недостаточно, приходится использовать программный код. Аналоги плохо поддерживают русский язык, а ближайший из них по функциональности к мультиагентной СДМС – G2 обладает высокой стоимостью. К достоинствам пакетов AnyLogic и G2 можно отнести использование ЯВУ, благодаря чему пакеты могут предоставлять разработчику моделей серьезный уровень функциональности.

Научное издание

Аксенов Константин Александрович
Гончарова Наталья Вадимовна

ГИБРИДНОЕ МОДЕЛИРОВАНИЕ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ

Монография



Технический редактор Кулакова Г.А.
Подписано в печать 19.12.2019
Бумага офсетная.
Гарнитура NewtonC
Формат 60×84 1/16
Печать трафаретная. Печ. л. 13,88.
Тираж 500 экз. Заказ № 031-19.

Отпечатано в типографии ИД «Академия Естествознания»,
440026, г. Пенза, ул. Лермонтова, 3