



**Уральский  
федеральный  
университет**

имени первого Президента  
России Б.Н.Ельцина

**Институт радиоэлектроники  
и информационных  
технологий**

**К. А. АКСЕНОВ  
Н. В. ГОНЧАРОВА**

# МОДЕЛИРОВАНИЕ И ПРИНЯТИЕ РЕШЕНИЙ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ

Часть 1

Учебное пособие

Министерство образования и науки Российской Федерации

Уральский федеральный университет  
имени первого Президента России Б. Н. Ельцина

К. А. Аксенов, Н. В. Гончарова

# **МОДЕЛИРОВАНИЕ И ПРИНЯТИЕ РЕШЕНИЙ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ**

## **Часть 1**

Рекомендовано Региональным отделением УрФО  
учебно-методического объединения вузов  
Российской Федерации по образованию  
в области радиотехники, электроники,  
биомедицинской техники и автоматизации  
в качестве учебного пособия  
для студентов высших учебных заведений,  
обучающихся по направлению подготовки  
220400 – Управление в технических системах в УрФО

Екатеринбург  
Издательство Уральского университета  
2015

УДК 681.5.01:005(075.8)  
ББК 32.965я73+65.050.2я73  
А42

Рецензенты:

Институт экономики УрО РАН, Центр экономической безопасности  
д-р физ.-мат. наук, проф. *А. Ф. Шориков* (Екатеринбург);  
канд. техн. наук, доц. *В. М. Кормышев* (директор ООО «УралСпорт-Сервис»)

Научный редактор – д-р техн. наук, проф. *Л. Г. Доросинский*

**Аксенов, К. А.**

А42 Моделирование и принятие решений в организационно-технических системах: учебное пособие. В 2 ч. Ч. 1 / К. А. Аксенов, Н. В. Гончарова. – Екатеринбург : Изд-во Урал. ун-та, 2015. – 104 с.

ISBN 978-5-7996-1321-1 (ч. 1)

ISBN 978-5-7996-1320-4

В учебном пособии отражены аспекты моделирования и принятия решений в организационно-технических системах. Основное внимание уделено моделированию бизнес-процессов с использованием автоматизированных средств поддержки принятия решений. Описаны процессы системного анализа, моделирования и принятия решений на промышленных предприятиях. Рассмотрены методы моделирования организационно-технических систем и существующие системы поддержки принятия решений. Пособие содержит примеры, иллюстрирующие материал. Предназначено для студентов дневной и дистанционной форм обучения направлений 230100 – Информатика и вычислительная техника, 220400 – Управление в технических системах, 080200 – Менеджмент специальности 080502 – Экономика и управление на предприятии (металлургия).

Библиогр.: 89 назв. Табл. 1. Рис. 17.

УДК 681.5.01:005(075.8)  
ББК 32.965я73+65.050.2я73

ISBN 978-5-7996-1321-1 (ч. 1)  
ISBN 978-5-7996-1320-4

© Уральский федеральный университет, 2015

## **ВВЕДЕНИЕ**

Учебное пособие посвящено 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.

Характерной особенностью решаемых сегодня задач является их многокритериальность, поэтому лицам, принимающим решения (ЛПР), приходится оценивать множество сил, влияний, интересов и последствий, характеризующих варианты решений.

Формализация процессов 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 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 ЛПР.

Повышение производительности и надежности, уменьшение стоимости и рисков, оценка чувствительности системы к изменениям параметров, оптимизация структуры — все эти проблемы встают как при эксплуатации существующих, так и при проектировании новых технических и организационных систем. Трудность понимания причинно-следственных зависимостей в слож-

ной системе приводит к неэффективной организации систем, ошибкам в их проектировании, большим затратам на устранение ошибок. Сегодня моделирование становится единственным практическим эффективным средством нахождения путей оптимального (либо приемлемого) решения проблем в сложных системах, средством поддержки принятия ответственных решений.

Применение ситуационных моделей в управлении способствует повышению эффективности принимаемых решений, сокращению времени принятия решений, повышению качества принимаемых решений, более рациональному использованию имеющихся ресурсов.

В данном учебном пособии 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 лицами, принимающими решения.

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., Виттих В. А., Гольдштейн С. Л., Емельянов С. В., Исидзука М., Карпов Ю. Г., Клыков Ю. И., Попов Э. В., Поспелов Д. А., Прицкер А., Ржевский Г. А., Скобелев П. О., 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.

**Идея учебного пособия** 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.*

Пособие состоит из введения, четырех глав, заключения, списка литературы.

В первой части пособия обоснована необходимость авто-

матизации процесса принятия решений (ППР), приведен обзор методов моделирования мультиагентных процессов преобразования ресурсов, рассмотрены системы, близкие по функциональности к системам динамического моделирования ситуаций (СДМС), и выполнен их сравнительный анализ, определены требования к СДМС мультиагентных процессов преобразования ресурсов.

Во второй части излагаются принципы построения СДМС мультиагентных процессов преобразования ресурсов и СППР технико-экономического проектирования, приведено описание данных систем, а также описаны принципы работы с ними.

Авторы благодарны Е. Ф. Смолий за оказанную неоценимую помощь при разработке и отладке СППР семейства VPsim. За предоставленную экспериментальную базу благодарим ООО «НПП «Системы автоматизации поддержки бизнеса».

Авторы также благодарят всех аспирантов и студентов ФГАОУ ВПО «УрФУ», принявших участие в отладке, проведении расчетов и оформлении результатов имитационного моделирования в процессе изучения курсов «Системы искусственного интеллекта», «Системы поддержки принятия решений», «Интерфейсы АСОИУ» и других дисциплин.

# ОБЗОР МЕТОДОВ И СРЕДСТВ СИТУАЦИОННОГО МОДЕЛИРОВАНИЯ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ (МППР)

## 1. СИТУАЦИОННОЕ УПРАВЛЕНИЕ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИМИ СИСТЕМАМИ (ОТС)

### *1.1. Ситуационный подход в управлении*

В становление теории мультиагентных процессов преобразования ресурсов и развитие систем поддержки принятия решений существенный вклад внесли работы Борщёва А.В. [1], Вавилова А.А. [2], Гольдштейна С.Л. [3–4], Емельянова С.В. [2], Исидзуки М. [5], Карпова Ю.Г. [6], Клыкова Ю.И. [7–9], Попова Э.В. [10], Поспелова Д.А. [11–12], Прицкера А. [13], Советова Б.Я. [14], Форрестера Дж. [15], Филипповича А.Ю. [16], Чистова В.П., Швецова А.Н. [17], Шеера А.В. [18], Уэно Х. [5], Яковлева С.А. [14], Jennings N.R. [19–21], Minsky M. [22], Wooldridge M.J. [19, 23] 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, содержащее указание на цель с желаемыми конкретными числовыми (в том числе временными) характеристиками [24].

Современная ОТС 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 [17].

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. [7–8] и Пospelов Д. А. [11–12] в своих ранних работах явно отождествляют ситуацию с состоянием. 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»* [12]. 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 [16].

Будем придерживаться следующего определения ситуации, данного в работе [16]: *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 [16] *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 [16]:

$$S_i : Q_j \xrightarrow{U_k} Q_i, \quad (1)$$

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 [16]: 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  $Q_i$ . Подобные правила преобразования называются *логико-трансформационными правилами* (ЛТП) или *корреляционными правилами*. Полный список ЛТП задает возможности системы управления воздействовать на процессы, протекающие в объекте. Очевидно, что в силу конечности числа различных воздействий все множество возможных полных ситуаций как-то распадается на  $n$  классов, каждому из которых будет соответствовать одно из возможных воздействий на объект управления.

Данное учебное пособие посвящено ситуационному моделированию дискретных процессов преобразования ресурсов.

## 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 [25–26].

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 [25–26].

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) [25]. На этой схеме прямоугольниками обозначены ресурсы предприятия, а овалами преобразователи ресурсов.

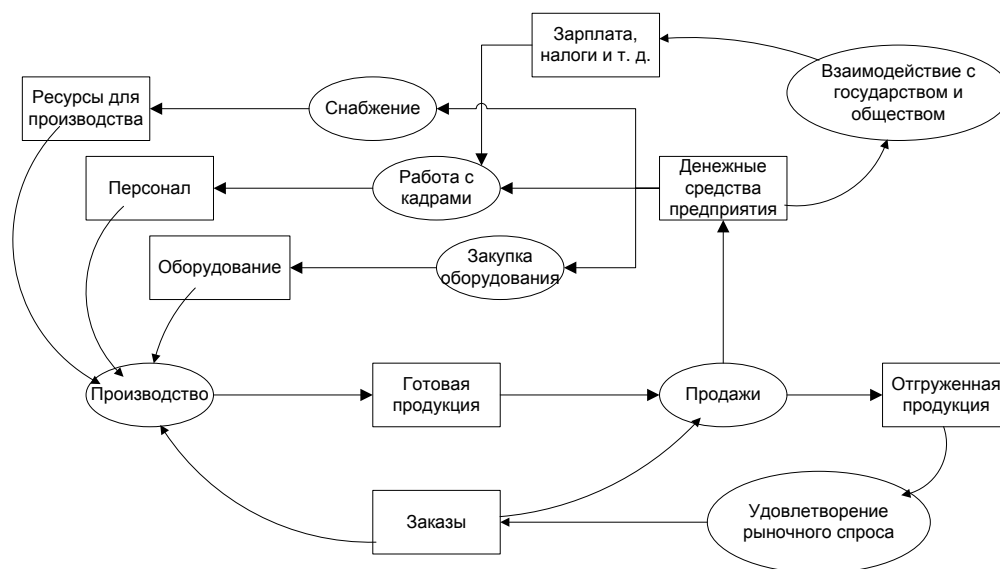


Рис. 1. Поточковая модель предприятия

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 авторов Клыкова Ю.И. и Пospelова Д.А. [8–9, 11–12] 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 [27]. Методы ситуационного управления нашли применение при автоматизации управления процесса капитального строительства [28–29] 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. 2).

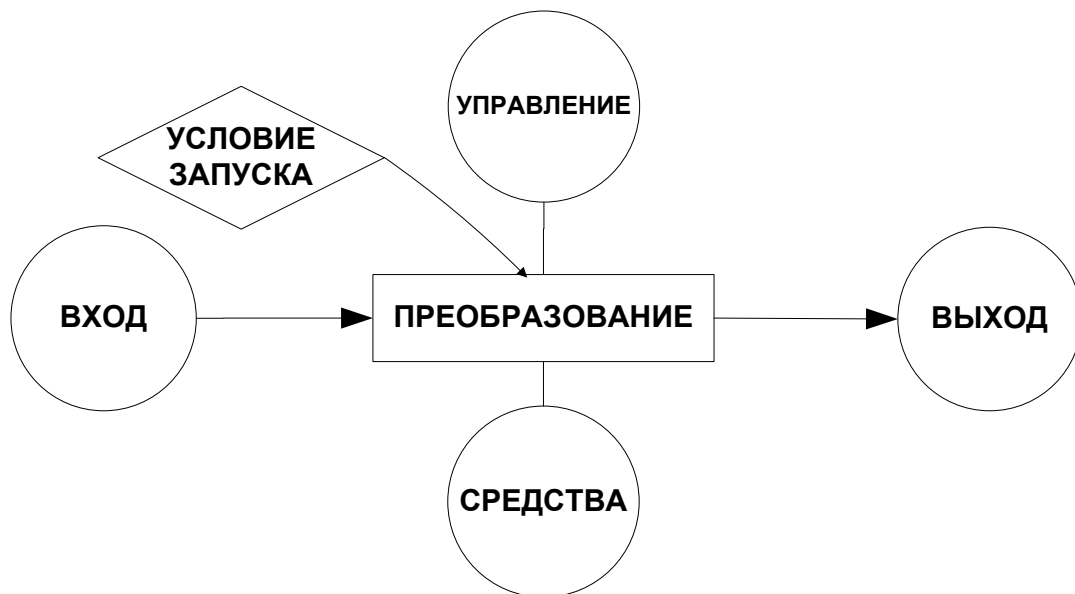


Рис. 2. Ситуационное представление процесса преобразования ресурсов

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 помо-

щью 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 [30] в процессе преобразования приведена на рис. 3.

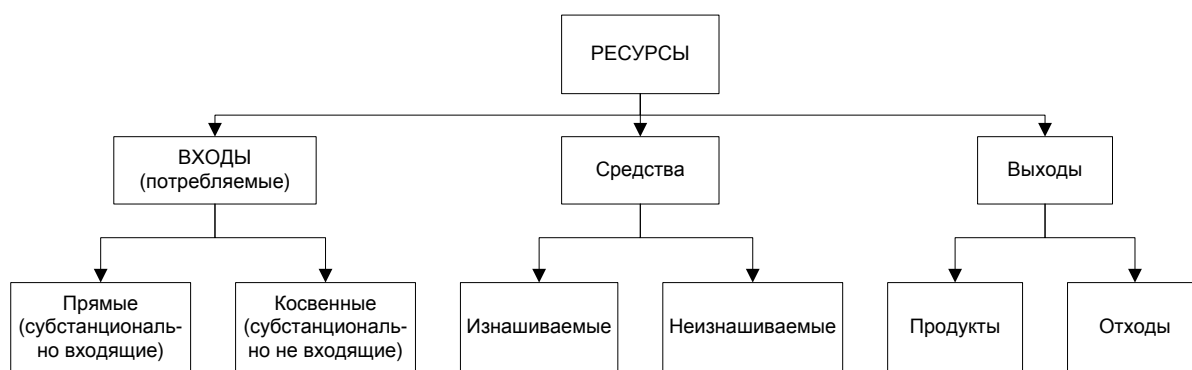


Рис. 3. Классификация ресурсов по типам использования

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 организационно-технических систем 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 [14]. Сложные процессы преобразо-

вания ресурсов, с точки зрения структурного подхода [2, 31–32], 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  $t$  [7].

Команда управления большой системой в момент времени  $t$  представляет собой совокупность команд, подаваемых на управляющие входы полюсников сети в момент времени  $t$ , и связей между командами, определяемых структурой сети. Команда управления в момент времени  $t$  соответствует макроситуации, которой принадлежит микроситуация  $s(t-1)$ . В общем случае полюсниками дискретной сети могут быть модели принятия решений. В этом случае командами управления полюсников служат указания на решение определенных задач из допустимого множества, а связи сети определяют порядок решения задач на заданном временном интервале. Таким образом, дискретная сеть

позволяет не только формализовать структуру коммуникационных связей между элементами объекта управления, но и описать процесс перехода объекта управления из одного состояния в другое. Функционирование дискретной сети выглядит внешне как смена ситуации на сети. Поскольку состояние дискретной сети может быть представлено в виде совокупности понятий и отношений между ними, то функционирование дискретной сети можно рассматривать как трансформацию понятийных структур, вершинами которых являются понятия, а ребрами служат отношения между ними [7]. 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, 13–16, 25–26].

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. [7].

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 от-

дельных 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 [7].

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 [25–26] был успешно применен аппарат системных графов высокого уровня интеграции, представленный в работе [2].

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.

## **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 [33–34].

В организационно-технических системах управления 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 [17, 35]. Отсюда вытекает потребность в интеллектуальной системе ППР, которая бы взяла на себя все формализованные функции исполнителей и оказала существенную поддержку при решении трудноформализуемых 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 весьма трудно описать алгоритмическими моделями; более адекватными оказываются модели представления знаний, позволяющие менять правила поведения и осуществлять логические выводы на основании содержания базы знаний (БЗ) [17].



В случае возникновения затруднений при решении задач должны быть проведены переговоры с людьми вышестоящего уровня, поставившим данную задачу и определившим критерии для её решения. При этом необходимо предоставить полную информацию о решаемой задаче с приведением возникших трудностей, их причин и возможными альтернативными вариантами решения. Данная процедура возможна также непосредственно при передаче задачи на нижний уровень, где она анализируется и где выдается заключение о возможности решения задачи в рамках данных установок. Передвижение задачи вниз по иерархии останавливается на определенном уровне. Данный уровень определяется требуемой степенью обобщенности информации, позволяющей кругу лиц, которые должны быть причастны к этому процессу, наиболее эффективно решать поставленные задачи [17].

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, способных обеспечить возможность определения сравнительной эффективности альтернативных вариантов с учетом широкого диапазона непредвиденных осложнений и изменения экзогенных факторов [35].

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 [17].

На уровне стратегического управления организационно-технической системы 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 [17].

На основании изложенного в данном разделе материала вытекают следующие требования к информационным технологиям ЛПР:

- 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;

- 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.

## 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 [9]. 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 [17, 33–34]. Различные определения данного термина обобщил Э.А. Трахтенгерц [34]: СППР – это система, выполняющая следующие функции:

- 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.

Согласование решений может осуществляться как до оценки возможных вариантов решения, сделанных соответствующим руководителем, так и после такой оценки. Генерацию возможных решений (сценариев) можно осуществить посредством программной реализации аналитических или имитационных моделей; с использованием ЭС; генерации сценариев путем комбинации различных операций, заданных руководителем или взятых из базы данных; используя подход, получивший название ситуационного управления [34].

В работе Клыкова Ю. И. [7] показано, что процесс описания структуры объектов в терминах единиц естественного языка и отношений между ними формально является процессом определения ситуаций. Таким образом, описание модели (обследование) проблемной области на ограниченном естественном языке существенно снижает требования к наличию знаний у аналитика (специалиста по извлечению знаний) в области программирования. Там же [7] показано, что структура дискретной сети является одновременно имитационной моделью структуры больших систем и языком описания ситуаций. В ранних работах Клыкова Ю. И., посвященных изложению принципов построения языка имитации, структура дискретной сети названа *базовой сферой знаний*.

*На основании вышеизложенного в СППР могут использоваться средства имитационного и ситуационного моделирования, экспертных систем.*

## 2.2. СППР в стратегическом управлении

При стратегическом управлении необходимость обрабатывать огромное количество внешней и внутренней информации требует разработки и внедрения в управление предприятием информационной системы, позволяющей руководителю [36]:

- 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 ScoreCard (BSC, система сбалансированных показателей) – мощный 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; 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 бизнес-процессов описан в [37].

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

$$BSC = \langle \text{Миссия, Виденье, Мероприятия,} \quad (2) \\ \text{Стратегии, Цели, Показатели, Связи} \rangle.$$

Методика системы сбалансированных показателей не позволяет количественно или качественно описать причинно-след-

ственные связи и применить какой-либо математический аппарат для решения задач стратегического управления, таким образом, в данной работе решается задача обеспечения стратегической методике 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 организационно-технической системе 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 вышеописанных 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 методике стратегического управления, например системы сбалансированных показателей.*

### **3. СИСТЕМНЫЙ АНАЛИЗ ОТС**

Данный раздел посвящен системному анализу (СА) ОТС на основе модели МППР. При проведении СА ОТС обычно описывают следующие составляющие: миссию, виденье, стратегии, внешние процессы, внутренние процессы (производственные процессы, бизнес-процессы и т. д.). Применение теории МППР позволяет по-новому взглянуть на ОТС с точки зрения динамических систем, основанных на знаниях, а позволяет также уделить внимание следующим элементам:

- моделям ЛПР, их знаниям, моделям поведения (процессам принятия решений);
- моделям координации и взаимодействия агентов;
- динамической составляющей процессов;
- рассмотрению отношений миссии, видения, стратегий, целей, КРІ и мероприятий (процессов) с помощью методике стратегического управления — системы сбалансированных показателей (ССП).

При построении иерархической модели на каждом уровне вводятся свои представления о системе и элементах. Элемент  $k$ -го уровня является системой для  $(k-1)$  уровня. Продвижение от уровня к уровню имеет строгую направленность, определяемую стратегией проектирования – дедуктивную нисходящую «сверху вниз» (top-down) или индуктивную восходящую «снизу вверх» (bottom-up) [38].

В контексте формализации процессов дедуктивную нисходящую стратегию проектирования используют в нотациях IDEF0, IDEF3, DFD [39–42], EPC [18], а также при построении иерархических моделей динамических процессов [2, 14] (агрегатах, сетях Петри, расширенных сетях Петри). Индуктивную восходящую стратегию используют в системных графах высокого уровня интеграции.

Для описания иерархической структуры МППР [36] были использованы системные графы высокого уровня интеграции (см. рис. 4) [2]:

$$\begin{aligned} \rightarrow \Sigma \\ PR_{L=i} = < \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=i}; \quad (3) \\ \{PR_{L=j}^{p_i}; p_i = 1, \dots, n_{L=j}^p\}_{j=2, \dots, i}; \{Relation_{AB}^{mk}\}_{L=i} > . \end{aligned}$$

Граф  $i$ -го уровня интеграции образуется в результате поэтапной интеграции графов  $\rightarrow \Sigma PR_1, PR_2, \dots, PR_{i-1}$  с образованием на каждом  $j$ -м этапе множества  $\rightarrow \Sigma \{PR_{L=j}^p; p=1, \dots, n_{L=j}^p\}$  процессов (подпроцессов)  $j$ -го уровня интеграции,  $L$  – уровень интеграции. Элементы множества мультиагентного процесса преобразования ресурсов  $\rightarrow \Sigma \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=i} \cup \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=i-1} \cup \dots \cup \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}$  и множества ресурсных отношений  $\rightarrow \Sigma \{Relation_{AB}^{mk}\}_{L=i} \cup \{Relation_{AB}^{mk}\}_{L=i-1} \cup \dots \cup \{Relation_{AB}^{mk}\}$  системного графа  $\circledast PR_{L=i}$  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  $\rightarrow \Sigma PR$  нулевого уровня интеграции, не вошедшие при поэтапной интеграции ни в один процесс  $PR_{L=j}^p$ .

С точки зрения динамического моделирования в имитации участвуют только те элементы, которые в результате при-

менения дедуктивной стратегии СА являются элементарными и в дальнейшем не детализируются. При использовании аппарата системных графов на первом шаге построения модели (0-й уровень интеграции) динамической системы получаем все необходимые данные для имитации.

В целом существующие подходы к проектированию сложных систем можно разделить на два больших класса [38]:

- *структурный (системный) подход* или анализ, основанный на идее алгоритмической декомпозиции, где каждый модуль системы выполняет один из важнейших этапов общего процесса;

- *объектный подход*, связанный с декомпозицией и выделением не процессов, а объектов, при этом каждый объект рассматривается как экземпляр определенного класса.

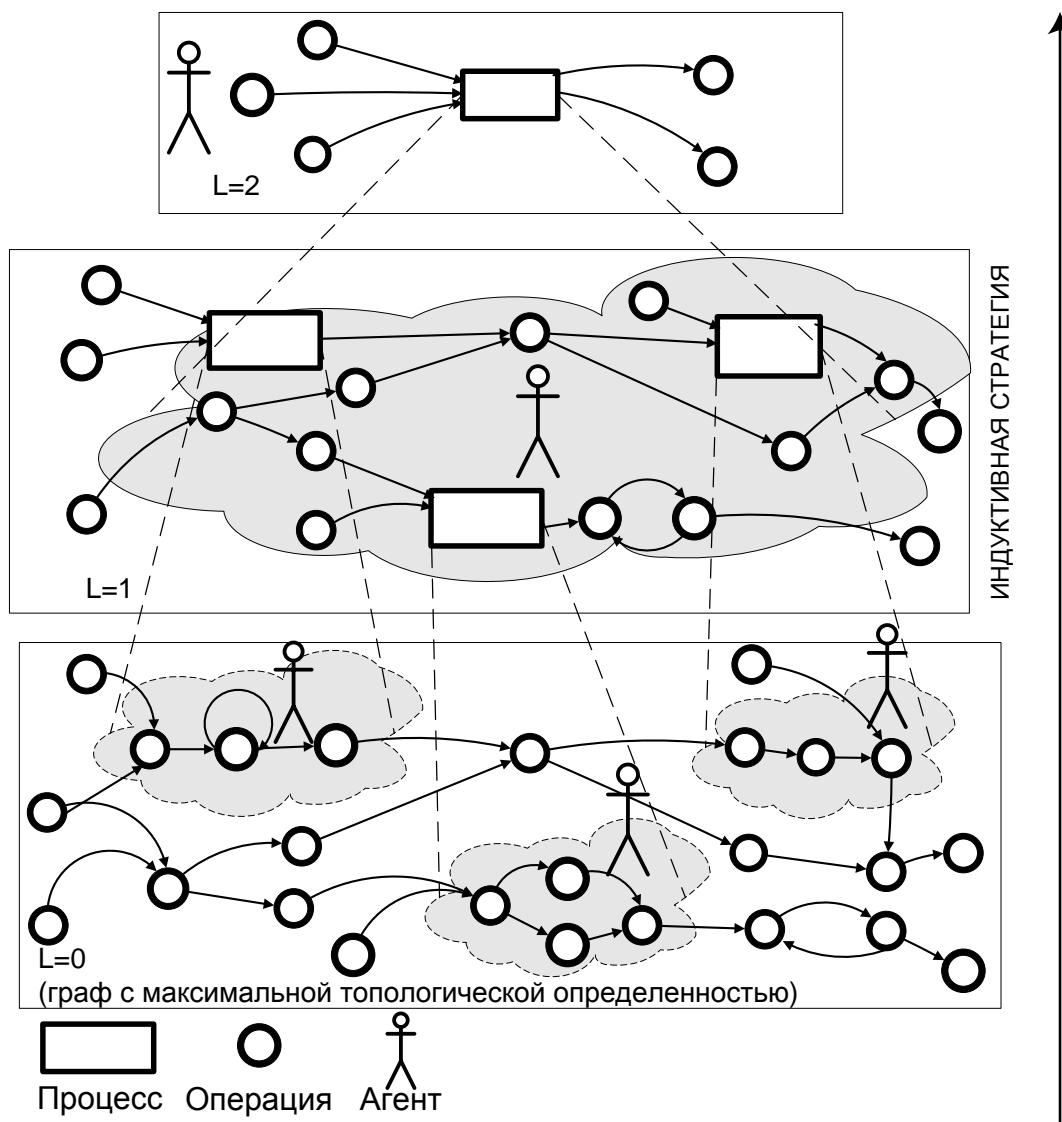


Рис. 4. Иерархическое представление МППР



Объектный (объектно-ориентированный подход (ООП), возникший как технология программирования больших программных продуктов, базируется на основных элементарных понятиях [44]:

- объекты и классы как объекты, связанные общностью структуры и свойств;
- классификация как средство упорядочения знаний;
- иерархия с наследованием свойств;
- инкапсуляция как средство ограничения доступа;
- методы и полиморфизм для определения функций и отношений.

ООП имеет систему условных обозначений и предлагает набор моделей для проектирования сложных систем. Широкое распространение объектно-ориентированных языков программирования C++, CLOS, Smalltalk, G2 и др. успешно демонстрируют жизнеспособность и перспективность этого подхода [38]. В последнее время этот подход успешно применяется и в CASE-средствах не только для проектирования программ, но и для моделирования БП.

Рис. 5 иллюстрирует концепцию дуальной стратегии проектирования при проектировании функциональной структуры для ЭС помощи специалиста по развитию мультисервисной сети связи (МСС).

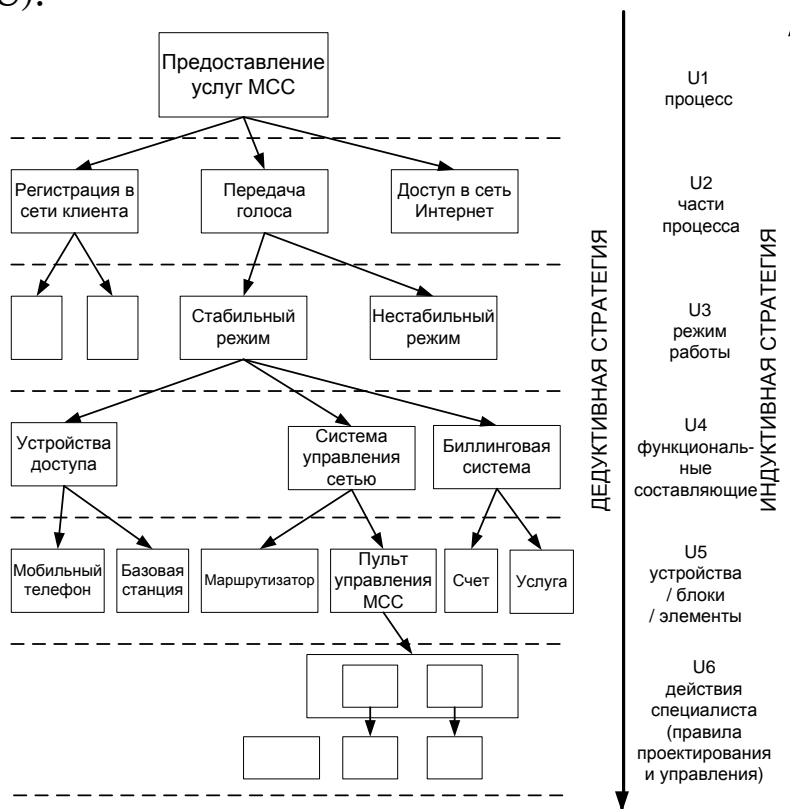


Рис. 5. Дуальная стратегия проектирования

Основанием для прекращения агрегирования и дезагрегирования является полное использование словаря терминов, которым владеет эксперт, при этом число уровней является значимым фактором успешности структурирования [120].

*В данной работе используется дуальная стратегия проектирования, модель МППР (и ее графическая нотация, в основе которой лежит метод системных графов высокого уровня интеграции), результаты которых программно реализованы в СППР семейства VPsim.* В следующем разделе также решается задача выбора модели представления знаний, которая наиболее соответствует решению задачи для предметной области МППР.

#### **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 (ПрО), 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 [15, 45]:

- неудачный выбор метода извлечения знаний, не соответствующего структуре предметной области;
- неадекватные модели и языки для представления знаний;
- отсутствие психологического контакта с экспертами ПрО;
- терминологическая рассогласованность языка экспертов и инженеров по знаниям;
- невозможность построения целостной модели ПрО в результате извлечения фрагментов знаний;
- потеря части знаний о ПрО в результате упрощения пространства эксперта;
- зависимость качества создаваемой системы от квалификации «посредников» между экспертами и инструментальными средствами — инженеров по знаниям;
- автономное использование методов 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 БЗ ИНС;
- необходимо преобразование результатов работы алгоритмов обучения на примерах в способ представления, поддерживаемый программными средствами 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 ПрО.

Сложность этапа структурирования знаний проявляется в том [17], что необходимо построить такую модель ПрО, 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 [46]. *Таким образом, минимизация усилий при переходе от модели ПрО к ее технической реализации является актуальной задачей.*

Следуя Н. Kangassalo [47], концепт рассматривается как фундаментальное понятие, описывающее именованный независимо идентифицируемый структурированный конструкт, объединяющий примитивы знаний. Концепт является интенциональной 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 [17].

#### 4.1. Анализ фреймовых моделей

В концептуальной модели ПрО необходимо описать структуру ПрО, определить поведение объектов и субъектов, существующих в этой структуре, построить логические модели их вза-

имодействия [17]. Minsky в своей работе [22] 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) [48].

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). Верхние уровни фрейма фиксированы и представляют сущности, всегда истинные в ситуации, описываемой данным фреймом. Нижние уровни заканчиваются слотами, которые заполняются конкретной информацией при вызове фрейма. Можно провести аналогию между фреймами и описанием процедур в языках программирования. Фрейм соответствует описанию процедуры, а означенный фрейм (фрейм-экземпляр) соответствует вызову процедуры. Отличие фреймов от описаний процедур состоит в том, что фреймы могут вызываться не по имени, а по соответствию текущей ситуации той ситуации, которую описывает данный фрейм. Кроме того, фрейм, слоты и механизм их означивания описывают ситуацию в семантических (а не синтаксических) терминах и в метатерминах. С каждым слотом фрейма связаны описания условий, которые должны быть соблюдены, чтобы могло произойти означивание слота. В простейших случаях эти условия могут сводиться к указанию семантических категорий, которым должно удовлетворять значение слота. В более сложных случаях условия могут касаться отношений между значениями, выбираемыми для нескольких слотов [10].

Преимущества подхода, основанного на фреймах, заключаются в следующем [17]: концепция фреймов естественным образом интегрируется с концептуальным моделированием пред-

метной области; структуры фреймов хорошо описываются средствами объектно-ориентированного проектирования; эффективно поддерживаются возможности наследования; обеспечивается иерархическое представление ПрО. Таким образом, выбор фреймов обосновывает применение ООП и объектных языков программирования при разработке СТЭП МСС, минимизирует затраты на создание программного обеспечения.

Анализ работ, выполненный А. Н. Швецовым в [17], 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 БД 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.

#### 4.2. Фреймовый подход Швецова для построения концептуальной модели предметной области (КМПО)

Для построения КМПО и решения задачи сокращения затрат на разработку ПО в работе использован фреймовый подход Швецова [17], основанный на совмещении фреймоподобных структур с конструкциями концептуальных графов J. F. Sowa [49–51]. Преимуществами данного подхода являются деление на активные и пассивные фреймы и учет поведения объекта.

Основная конструкция фрейм-концепта (ФК) представлена на рис. 6. 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 (ССцП), в которую включен блок выбора сценария

(БВСЦ), позволяющий формировать альтернативные пути поведения данного фрейма.

Фрейм-концепт (ФК)	
Имя фрейма (ИФ)	...
Тип фрейма (ТФ)	...
Информация о применении	...
Структура сценариев поведения (ССцП)	
	Блок выбора сценария (БВСЦ)
	Сценарий 1 (СЦ1)
	Сценарий 2 (СЦ2)
	..... .....
Структура слотов (ССЛ)	
	Слот 1
	...
	...

Рис. 6. Конструкция фрейма-концепта

Структура слотов (ССЛ) представляет собой совокупность двух структур: структуры концептов (СК) и структуры атрибутов (СТА) (см. рис. 7). 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<sub>*i*</sub> к ФК, где ИК<sub>*i*</sub> – имя *i*-го концепта. Для установления логической организации ПрО 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). Таким образом, Швецовым предлагается использовать фреймово-семантическое представление знаний.

Модель фрейм-концепта определяется следующим образом:  
 $ФК = \langle ИФ, ТФ, ИП, ССцП, ССЛ \rangle,$  (4)

$ССЛ = \langle СК, СТА \rangle,$  (5)

$СК = \{(ИК_1, КО_1), (ИК_2, КО_2), \dots, (ИК_n, КО_n)\},$  (6)

$СТА = \{(ИАт_1, МО_1, ЗА_1), (ИАт_2, МО_2, ЗА_2), \dots, (ИАт_m, МО_m, ЗА_m)\}.$  (7)

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



Рис. 7. Структура слотов

Таким образом, применение подхода к описанию ПрО в виде ФК и КГ, предложенное Швецовым, позволяет использовать фреймово-семантическую модель представления знаний. Задача перехода (совмещения) модели представления знаний, КМПО и их технической реализации на уровне БД решена в СППР *VPsim.MSS* [36, 52–53].

При построении ИНС автоматизации проектирования МППР еще одной из задач является построение механизма или языка вывода по БЗ ПрО. Данная задача решается в следующем разделе.

#### 4.3. Применение диаграмм последовательности для визуализации вывода на сети фрейм-концептов и концептуальных графах (диаграммы поиска решений)

В качестве средства формализации знаний МППР используется подход на основе ФК и КГ, предложенный Швецовым А. Н. [17] и реализованный применительно к промышлен-

ной СУБД MS SQL Server в виде оболочки ЭС «Конструктор фреймовых систем» (КФС) [36, 52], в последующем доведенной до системы VPsim.MSN.

Роль ООП в программировании всем известна, он всеми признан как весьма эффективный, и сейчас делаются попытки распространить его на другие области. Парадоксально, что подход, возникший для решения задач моделирования (язык Simula-67), именно в этой области долгие годы и не был востребован, и только сейчас начинает достаточно робко воплощаться в реальные разработки (Omola, Dymola, Model Vision, AnyLogic, G2). В моделировании этот подход называют объектно-ориентированным моделированием (ООМ). Основные компоненты, составляющие UML, включают описание семантики UML, его графической нотации и дополнительных понятий, позволяющих расширить смысл основных понятий языка.

При проведении СА в качестве основы описания структуры ФК может быть использована диаграмма классов языка UML. Дальнейшее описание КГ (семантики) и наполнение данными КМПО образует базу знаний.

Для реализации визуального конструктора механизма логического вывода КФС предложено использование диаграмм последовательности языка UML. Диаграмма последовательности графически описывает последовательность вызова методов между классами при решении определенной задачи (сценария). Данный подход позволяет визуально (в виде блок-схемы) описать ход решения задачи — последовательность вызовов процедур (методов \ демонов) от одного фрейма к другому.

Согласно правилу чтения диаграммы последовательности выполнения методов указывается сверху вниз. Для организации циклов и возможности перебора альтернатив методы усовершенствованной диаграммы последовательности расширены возможностью отката (перехода) на ранний шаг (метод). Метод может реализовывать функцию выбора (означивания) значения свойства фрейма-экземпляра или фрейма-класса (в случае множественного выбора). Таким образом, применение диаграммы последовательности позволяет описывать сценарий решения задачи, в рамках которого путем вызова методов реализуется обработка фреймов. По аналогичным принципам строятся диалоговые ЭС.



Пример диаграммы последовательности представлен на рис. 8.

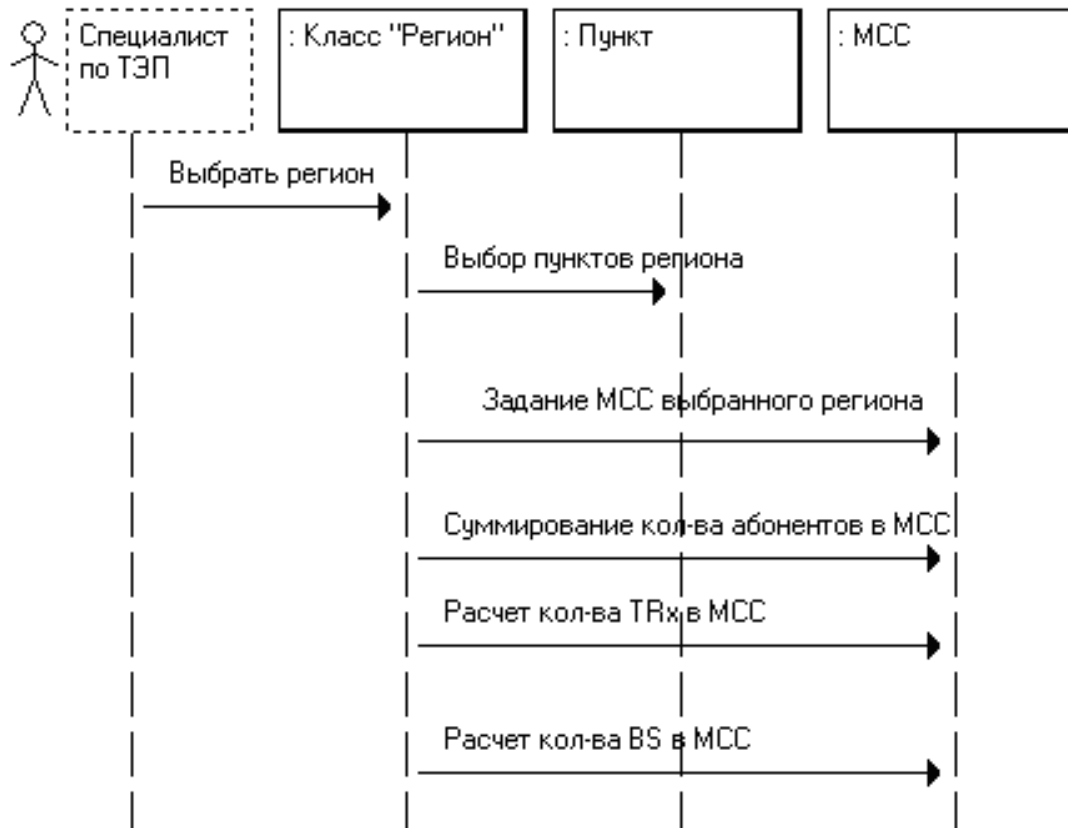


Рис. 8. Диаграмма «Поиск решения»:

МСС – мультисервисная сеть связи; TRx – приемо-передатчик;  
BS – базовая станция

Таким образом, с помощью данного подхода реализован визуальный объектно-ориентированный конструктор построения онтологии и построения механизма вывода на знаниях, что в совокупности с расширением графической нотации языка UML позволит реализовать диалоговую ЭС. Данный конструктор, при условии наполнения знаниями о предметной области МСС и правилами технико-экономического проектирования (ТЭП), представляет собой интеллектуальную систему автоматизированного проектирования МСС.

## 5. МЕТОДЫ МОДЕЛИРОВАНИЯ

В разделе приводится обзор существующих методов применяемых при решении задач анализа, моделирования и принятия решений в предметных областях процессов преобразования ресурсов и ОТС.

## 5.1. Имитационное моделирование

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

*Имитационное моделирование* — это разработка компьютерных моделей и постановка экспериментов на них. Целью моделирования в конечном счете является принятие адекватных (т.е. обоснованных, целесообразных и реализуемых) управленческих решений. Компьютерное моделирование становится сегодня обязательным этапом в принятии ответственных решений во всех областях деятельности человека в связи с усложнением систем, в рамках которых человек должен действовать и которыми он должен управлять. Знание концепций, принципов и возможностей ИМ, умение строить и использовать модели являются необходимыми требованиями к инженеру, менеджеру, бизнес-аналитику [6].

В английском языке для обозначения процесса моделирования используются два различных слова: *modeling* и *simulate*. При этом первому слову соответствует процесс проектирования, создания модели устройства или предметной области. Под имитацией (*simulate*) понимают исследование (испытание, прогонку) модели. Процесс имитации невозможен без предварительного создания модели. В свою очередь последующая имитация накладывает ограничения на язык и способы описания модели. Исходя из вышесказанного ИМ можно рассматривать как один из способов моделирования, который предполагает дальнейшее исследование. Отличительной чертой неимитационных моделей является их статичность, поэтому к ним можно отнести языки описания декларативных знаний или данных. Примерами могут являться фактографические системы (базы данных) и модели БД (реляционная, объектная), CASE системы проектирования ERwin и Rational Rose и соответствующие им модели и языки IDEFX1, UML [16].

Имитационное моделирование может применяться в самых различных сферах деятельности. Ниже приведен список задач, при решении которых моделирование особенно эффективно [54]:

- 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* [13–14], *дискретно-событийное* моделирование [2, 13–14, 54], *системная динамика* [15] и *агентное* моделирование [1; 6; 17; 36]. 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 [36; 55].

Теоретической базой создания средств ИМ являются широко распространенные математические схемы описания динамических процессов (расширенные сети Петри [14], системы массового обслуживания [14; 56], модели системной динамики [15]). Новый подход к моделированию динамических процессов, к которым относятся цепочки поставок (логистика), технологические, производственные, организационные и бизнес-процессы, предлагает концепция процессов преобразования ресурсов [25; 36; 55], 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 произ-

водственно-экономических системах, представлены в [16; 36; 55]. К распространенным в настоящее время проблемно-ориентированным СИМ в области дискретных процессов преобразования ресурсов относятся следующие: AnyLogic [1; 6; 57–59], Arena [60], BPsim.MAS [36; 55; 62], ARIS [4; 18], ReThink [10; 61].

## 5.2. Экспертное моделирование

В работе [16] выделен класс систем экспертного моделирования, которые имитируют процессы рассуждения человека. Исходными данными для них являются декларативные и процедурные знания, поэтому их также называют *системами, основанными на знаниях (knowledge-based system)*, 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 [14, 16].

ЭС — наиболее распространенный класс ИС, 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 [63].

Существует множество различных определений экспертной системы, при этом в большинстве случаев ее структура остается типовой и может включать следующие компоненты: *база знаний; база данных; машина вывода; интерфейс с пользователем; модуль извлечения знаний и обучения; компонент приобретения и объяснения знаний* [10; 16], 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* фиксирует шаги, 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 представляет собой *вывод на знаниях*. Механизм вывода во многом зависит от используемого языка представления знаний и может быть логическим, нечетким, вероятностным, продукционным и т. д. К экспертному моделированию относятся методы [16]:

- формирования, изменения, дополнения и оптимизации БЗ;
- обучения, извлечения и объяснения ЭС;
- ведения диалога и разработки интерфейса взаимодействия;
- описания предметной области на ЯПЗ;
- разработки алгоритмов и стратегий вывода;
- прогнозирования, экстраполяции и эвристического анализа;
- интеграции ЭС с другими системами.

Если при моделировании (выводе) ЭС опирается на исходные данные и на основании правил, хранимых в БЗ, полу-

чает результат, то такой вывод называется *прямым*. Если ЭС осуществляет поиск всех возможных комбинаций исходных данных, приводящих к одному (заданному) результату, то такой вывод называется *обратным*.

Большинство ЯПЗ можно представить в виде сетевой структуры (семантические сети, фреймы, сценарии, продукции), поэтому в них активно используются графовые методы поиска (в ширину, в глубину и др.)

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 [16; 25; 36; 45; 55].

### 5.3. Ситуационное моделирование

Одним из перспективных направлений создания моделей принятия решений, позволяющих 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 XX в. 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 [4].

Для описания ситуаций используются *семиотические (ситуационные) языки и модели*, среди которых можно выделить следующие основные подходы [16]:

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

Дискретная ситуационная сеть представляет собой сложную семантическую сеть [9]. Каждая ситуация описывается ориентированным графом (сетью), а для представления вложенности («ситуации ситуаций») используются гиперграфы, где некоторый фрагмент семантической сети определяет ситуацию и может рассматриваться как одна вершина сети.

RX-коды представляют собой язык бинарных отношений и имеют в качестве ядерной конструкции запись следующего вида [64]:

$$x_1 = x_2 r_2 x_3 r_3, \quad (8)$$

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 [10], фреймов [5; 10; 22; 65] и продукций [10; 48].

В [16] отмечается, что методы представления знаний в ситуационных системах и экспертных системах аналогичны. Еще больше они сблизилась 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.» [9].

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.

#### 5.4. Мультиагентный подход

Для решения задачи построения моделей ЛПР на разных уровнях сложной системы (см. раздел 2) целесообразно использовать теорию мультиагентных систем – 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 [66], проектирование [67] и т.п. Важной областью применения мультиагентных технологий является моделирование. В этой области Д.А. Поспелов [67] выделяет два класса задач. К первому классу он относит 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. В задачах второго класса агенты самостоятельно решают свои локальные задачи, используя общие, как правило, ограниченные ресурсы [66].

Термин «агент» происходит от латинского глагола *agere*, что означает «действовать», «двигать», «править», «управлять» [68]. Это определение правильно выражает суть современных интеллектуальных компьютерных агентов, которые могут функционировать автономно от имени своего владельца (человека-пользователя или другой вычислительной системы), и позволяет решать самые разнообразные задачи по обработке информации. Для успешной работы агент должен обладать достаточными интеллектуальными способностями, чтобы в сфере своих задач замещать действия человека-владельца, должен иметь возможности взаимодействия с владельцем или пользователем для получения соответствующих заданий и передачи результатов, должен ориентироваться в среде своего существования и принимать необходимые решения [17].

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 интел-



лектуальными способностями [69–70]. В дальнейшем будем придерживаться данного определения.

В [69] обсуждается пример применения мультиагентной системы (МАС) для планирования работы гибкой производственной системы. К преимуществам обсуждаемой мультиагентной системы планирования отнесены следующие: 1) формализация точек принятия решений (сценариев обработки отдельных ситуаций) в виде агентов, что по сути своей относится к этапу формализации знаний; 2) планировщик «встраивается» динамично (работает в реальном времени) посредством взаимодействий (переговоров) между отдельными элементами МАС и тем самым готов изменять план в случае задержек или неожиданных (внештатных) ситуаций; 3) сеть агентов, связанная отношениями, самостоятельно координирует свои действия.

Дополнительное преимущество от мультиагентного планирования — возможность автоматического информирования участников процесса об изменениях на объекте управления, что дает прозрачность управления. В процессе разработки и внедрения МАС планирования происходит формализация знаний о предметной области и автоматизируется процесс принятия решений, тем самым облегчается деятельность, связанная с принятием решений [71].

В России наиболее значительные результаты в исследовании МАС получены В. И. Городецким, применение данных результатов иллюстрируется на примерах решения следующих задач [76–82]:

- управление взаимодействием в P2P-сетях;
- автономное управление воздушным движением;
- управление ресурсами грид;
- самоконфигурации оверлейных сетей;
- автономные операции подводных роботов.

К наиболее значительным практическим результатам применения МАС можно отнести развитие и применение на практике аппарата сетей потребностей и возможностей (ПВ-сетей) [72–75]. Данный подход придерживается «классического» понимания мультиагентной системы и ориентирован на решение задач в вычислительных сетях [71].

Аппарат ПВ-сетей, разработанный В. А. Виттихом, П. О. Скобелевым, Г. А. Ржевским, программно реализован

в виде технологии MAGENTA, которая нашла применение в семействе прикладных интеллектуальных систем планирования для следующих объектов [72–75]: предприятия, управляющего океанским танкерным флотом; транспортного предприятия, занимающегося перевозками грузов; организации, деятельность которой связана с управлением проектами.

Две базовые характеристики — автономность и целенаправленность — позволяют отличать интеллектуального агента от других программных и аппаратных объектов (модулей, подпрограмм, процедур и т. п.). Интеллектуальным агентам присущи следующие основные свойства [17; 66]:

- автономность — способность функционировать без вмешательства со стороны своего владельца и осуществлять контроль собственных действий и внутреннего состояния. Автономность предполагает относительную независимость агента от окружающей среды, т. е. наличие «свободы воли», обуславливающей собственное поведение, которое должно быть обеспечено необходимыми ресурсами;
- активность — способность к организации и реализации действий;
- общительность — взаимодействие и коммуникация с другими агентами;
- реактивность — адекватное восприятие состояния среды и реакция на его изменение (соответствует рефлексорному поведению животного);
- целенаправленность, предполагающая наличие собственных источников мотивации;
- наличие базовых знаний о себе, о других агентах и об окружающей среде;
- убеждения — переменная часть базовых знаний, меняющихся во времени;
- желания — стремление к определенным состояниям;
- намерения — действия, которые планируются агентом для выполнения своих обязательств и/или желаний;
- обязательства — задачи, которые выполняет один агент по просьбе и/или поручению других агентов.

Интеллектуальная мультиагентная система представляет собой множество интеллектуальных агентов, распределенных в сети, которые мигрируют по ней в поисках релевантных дан-

ных, знаний, процедур и кооперируются для достижения поставленных перед ними целей. В мультиагентных системах (МАС) множество автономных агентов действуют в интересах различных пользователей и взаимодействуют между собой в процессе решения определенных задач. Примерами таких задач являются: управление информационными потоками и сетями, управление воздушным движением, поиск информации в сети Интернет, электронная коммерция, обучение, электронные библиотеки, коллективное принятие многокритериальных управленческих решений и другие [66].

Основными направлениями научного поиска являются [17] теории агентов, которые рассматривают математические методы и формализмы абстрактного представления структуры и свойств агентов и способы построения рассуждений в таких формальных 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.

Главная черта МАС, отличающая их от других интеллектуальных систем, — взаимодействие между агентами. Взаимодействие означает установление двусторонних и многосторонних динамических отношений между субъектами. Оно является не только следствием деятельности агентов, но и необходимым условием формирования виртуальных сообществ. К базовым видам взаимодействия между агентами относятся [66]: *кооперация* (сотрудничество); *конкуренция* (конфронтация, конфликт); *компромисс* (учет интересов других агентов); *конформизм* (отказ от своих интересов в пользу других); *уклонение от взаимодействия*.

Взаимодействие агентов обусловлено целым рядом причин, важнейшими среди которых являются следующие [66]:

*Совместимость целей (общая цель)*. Эта причина обычно порождает взаимодействие по типу кооперации. Несовместимость целей или убеждений обычно порождает конфликты, позитивная роль которых заключается в стимулировании процессов развития.

*Отношение к ресурсам*. Ресурсами будем называть любые средства, используемые для достижения агентами своих целей. Ограниченность ресурсов, которые используются многими агента-

ми, обычно порождает конфликты. Одним из самых простых и эффективных способов разрешения подобных конфликтов является *право сильного* — сильный агент отбирает ресурсы у слабых.

*Необходимость привлечения недостающего опыта.* Каждый агент обладает ограниченным набором знаний, необходимых ему для реализации собственных и общих целей. В связи с этим ему приходится взаимодействовать с другими агентами. При этом возможны различные ситуации: а) агент способен выполнить задачу самостоятельно; б) агент может обойтись без посторонней помощи, но кооперация позволит решить задачу более эффективным способом; в) агент не способен решить задачу в одиночку. В зависимости от ситуации агенты выбирают тип взаимодействия и могут проявлять разную степень заинтересованности в сотрудничестве.

*Взаимные обязательства.* Обязательства являются одним из инструментов, позволяющих упорядочить хаотические взаимодействия агентов. Они позволяют предвидеть поведение других агентов, прогнозировать будущее и планировать собственные действия. Формальное представление целей, обязательств, желаний и намерений, а также всех остальных характеристик составляет основу ментальной модели интеллектуального агента, которая обеспечивает его мотивированное поведение в автономном режиме.

Перечисленные причины в различных сочетаниях могут приводить к разным формам взаимодействия между агентами, например:

- простое сотрудничество, которое предполагает интеграцию опыта отдельных агентов (распределение задач, обмен знаниями и т. п.) без специальных мер по координации их действий;
- координируемое сотрудничество, когда агенты вынуждены согласовывать свои действия для того, чтобы эффективно использовать ресурсы и собственный опыт;
- непродуктивное сотрудничество, когда агенты совместно используют ресурсы или решают общую проблему, не обмениваясь опытом и мешая друг другу.

Коллективное поведение агентов в МАС предполагает кооперацию агентов при коллективном решении задач. В процессе работы мультиагентной системы агент может обращаться за помощью к другим агентам, если не в состоянии решить постав-

ленную перед ним задачу самостоятельно. При этом агенты могут строить планы совместных действий, не только полагаясь на свои возможности, но и анализируя планы и намерения и других членов коллектива. Моделирование коллективного поведения необходимо также в случаях, когда агенты для решения своих задач используют общий ограниченный ресурс. Каждый агент вынужден учитывать наличие других агентов, а выбор стратегии действий одного агента обычно зависит от поведения остальных.

Известные подходы проектирования агентно-ориентированных систем можно разделить на две группы [17]:

- 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, поддерживающих объектно-ориентированные методы разработки ИС, среди которых наиболее известными являются All Fusion фирмы Computer Associate и Rational фирмы IBM, процесс проектирования в которых основывается на языке объектно-ориентированного проектирования UML. Современное средство имитационного моделирования AnyLogic, поддерживающее агентный подход, использует расширение языка UML-RT [57–59, 83]; подробнее данный инструментальный рассматривается в разделе 6.

Вторая группа методологий строится на расширении традиционных методов инженерии знаний [17]. 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 выделить следующие требования к функциональным возможностям СДМС в предметной области процессов преобразования ресурсов:

- 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 следующем разделе ставится и решается задача разработки модели МППР, учитывающей вышеуказанные требования за счет интеграции следующих математических аппаратов: имитационного моделирования (дискретных процессов преобразования ресурсов), экспертных систем, ситуационного и мультиагентного моделирования.

### *5.5. Модель мультиагентных процессов преобразования ресурсов (МППР)*

В разделе рассматривается предметная область МППР [25–26, 36, 55], охватывающая такие классы процессов, как производственные, технологические, организационные, бизнес-процессы (БП) и цепочки поставок, и рассматривается возможность применения данного подхода для решения задачи технико-экономического проектирования (ТЭП) и бизнес-моделирования мультисервисных сетей связи (МСС).

Процесс выхода оператора в новый регион в концепции МППР можно представить в следующем виде: вход (состояние региона: политическое, географическое, климатическое, перспектива развития региона, состояние рынка связи); объем инвестиций; состояние рынка поставщиков оборудования и технологий; оборудование и технологии, используемые в процессе развертывания и эксплуатации МСС); выход (технико-экономический проект МСС, новая МСС с ее показателями, предоставляемые услуги, отдача инвестиций, новые клиенты); средства (технологии, оборудование, сотрудники, подрядчики); условия запуска (закон о связи, лицензия); агенты – ЛПР (учредители, техн. специалисты, маркетологи).



МППР реализована в результате интеграции ИМ, экспертного и ситуационного моделирования к области процессов преобразования ресурсов [36, 55]. МППР описывается следующей кортежной модели [36]:

$$M = \langle Name, desc, O, \{Relation\}, A^{self} \rangle, \quad (9)$$

где *Name* – имя модели; *desc* – описание модели; *O* – объекты (элементы), ресурсы, средства, преобразователи, сигналы, заявки, цели, параметры, агенты, сообщения; *Relation* – связи; *A<sup>self</sup>* – собственные атрибуты модели.

Модель МППР [36; 55] является развитием модели, предложенной Аксеновым К. А. и Клебановым Б. И. [25–26]. Основными объектами модели МППР являются (рис. 10): операции (*Op*), ресурсы (*RES*), команды управления (*U*), средства (*MECH*), процессы (*PR*), источники (*Sender*) и приемники ресурсов (*Receiver*), перекрестки (*Junction*), параметры (*P*), агенты (*Agent*). Описание причинно-следственных связей между элементами преобразования и ресурсами задается объектом «связь» (*Relation*). Существование агентов предполагает наличие ситуаций (*Situation*) и решений (планов действий) (*Decision*) [36].

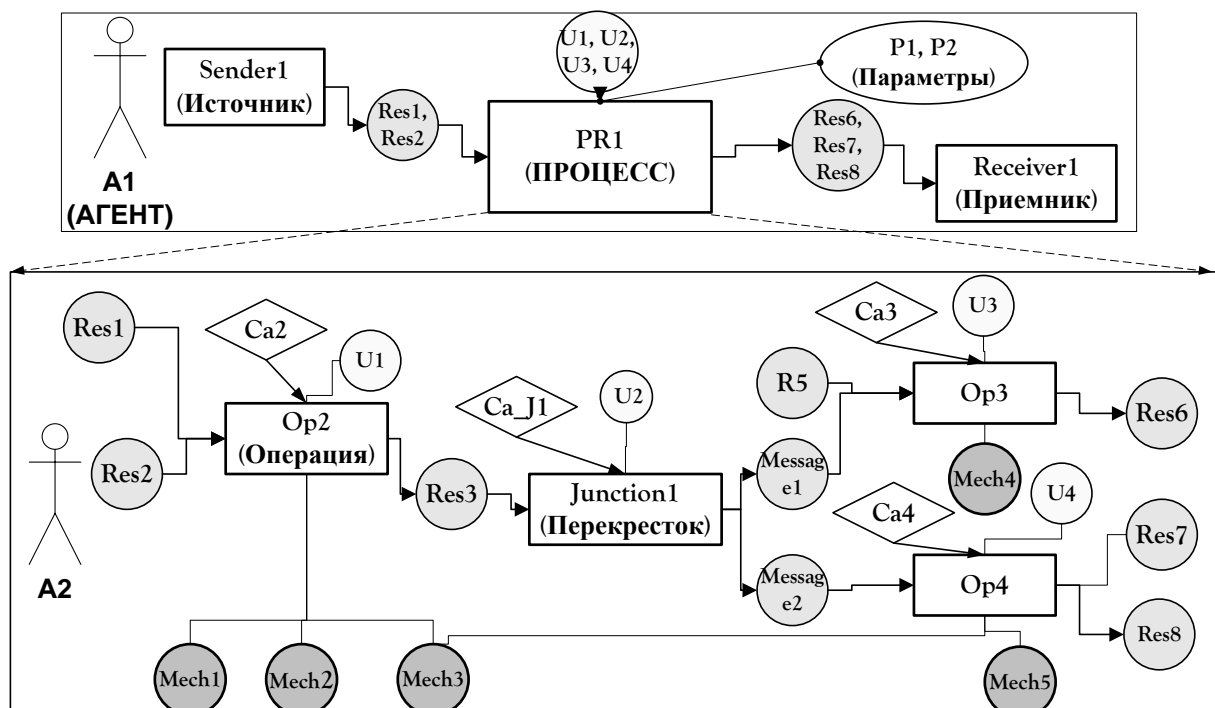


Рис. 10. Объекты модели МППР

Агенты управляют объектами процесса преобразования ресурса. Агент выполняет следующие действия (рис. 11): 1) анализирует текущую ситуацию; 2) диагностирует ситуацию, обра-



щается к БЗ. В случае определения соответствующей ситуации агент пытается найти решение (сценарий действий) в БЗ или выработать его самостоятельно; 3) вырабатывает (принимает) решение; 4) определяет (переопределяет) цели; 5) контролирует достижение целей; 6) делегирует цели своим и чужим объектам процесса преобразования ресурсов, а также другим агентам; 7) обменивается сообщениями.

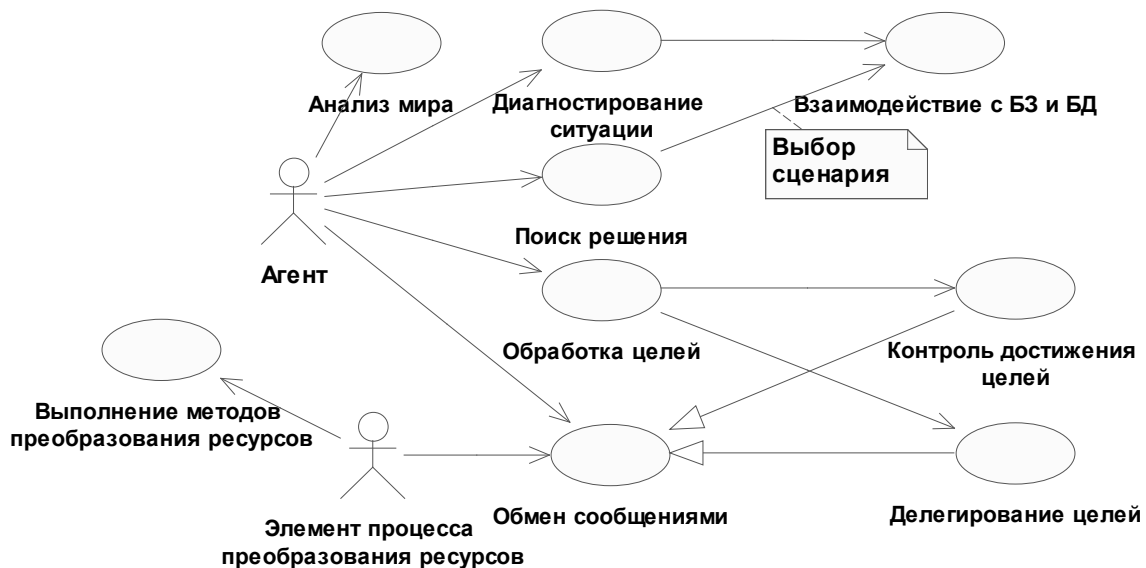


Рис. 11. Диаграмма прецедентов, определяющая отношения между агентом и элементом процесса преобразования ресурсов

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

Модель интеллектуального агента следующего вида [36]:  
 $Agent = \langle Name, G\_Ag, prior, KB\_Ag, M\_In, M\_Out, SPA, Control\_O, AU, AD \rangle,$  (10)

где  $Name$  – имя агента;  $G\_Ag$  – цели агента;  $prior$  – приоритет агента;  $KB\_Ag$  – база знаний агента;  $M\_In$  – количество входящих сообщений;  $M\_Out$  – количество исходящих сообщений;  $SPA$  – сценарии поведения;  $Control\_O$  – множество управляемых объектов процесса преобразования ресурсов;  $AU$  – множество агентов «начальников»;  $AD$  – множество агентов подчиненных.

Для построения ядра моделирующей системы был использован аппарат продукционных систем [36]. Структура продукционной системы МППР следующая:

$$PS = \langle Rps, Bps, Ips \rangle, \quad (11)$$

где  $Rps = \{RES(t)\} \cup \{MECH(t)\} \cup \{Order(t)\} \cup$   
 $\cup \{Sig(t)\} \cup \{Message(t)\} \cup \{U(t)\} \cup \{G(t)\} -$

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.

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

$$RULE_k^{OP} = \langle C_a(t), A_{IN}(t_{C_a}), A_{Lock}(t_{Lock}), A_{UnLock}(t_{UnLock}), A_{OUT}(t_{End}),$$

$$Status^{RULE}, time_{RULE}, prior, kind\_prior, break\_off \rangle, \quad (12)$$

$$A_{IN}(t_{C_a}) = \langle A_{in}^{RES}(t_{C_a}), A_{in}^{MECH}(t_{C_a}) \rangle, \quad (13)$$

$$A_{Lock}(t_{Lock}) = \langle A_{Lock}^{MECH}(t_{Lock}), A_{Lock}^{RES}(t_{Lock}) \rangle, \quad (14)$$

$$A_{UnLock}(t_{UnLock}) = \langle A_{UnLock}^{RES}(t_{UnLock}), A_{UnLock}^{MECH}(t_{UnLock}) \rangle, \quad (15)$$

$$A_{OUT}(t_{End}) = \langle A_{in}^{RES}(t_{End}), A_{in}^{MECH}(t_{End}) \rangle, \quad (16)$$

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

Средство  $Mech_y$ , обладающее моделью поведения, дополнительно отнесено к правилу преобразования, которому соответствует следующая структура [36]:

$$RULE_y^{MECH} = \langle t_{Create}^{mech}, Status_{Mech}, Am^{in}, Am^{out}, Am^{use}, Am^{lock}, Am^{unlock}, Am^{other}, t_y^{other}, product_y \rangle, \quad (17)$$

где  $t_{Create}^{mech}$  — время создания;  $Status_{Mech} = \{wait, active, lock, other\}$  — 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).

За основу модели интеллектуального агента в МППР [36] взята обобщенная модель интеллектуального агента (ОМИА), предложенная Швецовым А.Н. Данная модель адаптирована к предметной области процессов преобразования ресурсов. В основе правил поведения агентов взят специализированный объектно-ориентированный язык *RADL* (Reticular Agent Definition Language). Применительно к предметной области процессов преобразования ресурсов структура правил поведения агентов выглядит следующим образом [36]:

**Name** <Имя правила>

**If** < Message Conditions, RCP Conditions, G\_Ag Conditions >

**Then** < G\_Ag Changes; Message Actions; Private Actions > ,

где *Message Conditions* — условия, относящиеся к сообщениям; *RCP Conditions* — условия, относящиеся к преобразователям и ресурсам (*RCP* — *Resources conversion process*); *G\_Ag Conditions* — условия, относящиеся к целям;

*G\_Ag Changes* – действия по изменению текущих целей агента; *Message Actions* – действия по формированию сообщений; *Private Actions* – действия (план действий), связанные с преобразователями и ресурсами, направленные на достижение поставленных целей. Части правил могут быть представлены в виде предикатов первого порядка.

Алгоритм работы агента состоит из следующих действий [36]:

**1. Анализ мира** 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 модельно-

му множеству для поиска пути к цели  $ga^j$  на глубину в  $gr$  шагов). Полагаем, что параметр  $gr$  определяется выбранной стратегией  $s^i$ , и вводим два счетчика: счетчик модельных шагов  $Smgr$  и счетчик актуальных шагов  $Scgr$ . Выполняем модельные шаги для модельных множеств на глубину  $gr$ . После каждого шага вычисляем предполагаемую оценку состояния МА-мира для всех моделируемых ИО (элементов процесса преобразования) и делаем следующий модельный шаг. Повторяем модельные шаги до достижения  $Smgr=gr$ . 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):

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

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.

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

Если ситуация не изменилась, то переходим к следующему шагу.

**Генерация (выработка) решения** (если решения в БЗ не найдены). 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 МА-мира и определяет отклонения от цели. Поскольку целей может быть несколько (в случае  $j > 1$ ),

необходимо сформировать комплексную оценку положения интеллектуального агента в пространстве целей

$$\Delta_k = \sum_{j=1}^n a_i \Delta(ga^j, g\hat{a}^j), \quad (18)$$

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

Проводим сравнение с допустимым комплексным отклонением  $\Delta d$  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  $\Delta_k \leq \Delta d$ , то продолжается функционирование интеллектуального агента в той же стратегии.

**4. Формирование модели мира** данного агента (прогнозирование состояния на следующем шаге)  $FV(t+1)$ . Формируем новое текущее состояние МА-мира 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. 12 приведена схема работы интеллектуального агента.

В качестве модели представления знаний в МППР используется фреймово-семантическое представление. Для построения концептуальной модели предметной области и решения задачи сокращения затрат на разработку программного обеспечения (ПО), в [36] использован подход, основанный на совмещении фреймоподобных структур с конструкциями концептуальных графов.

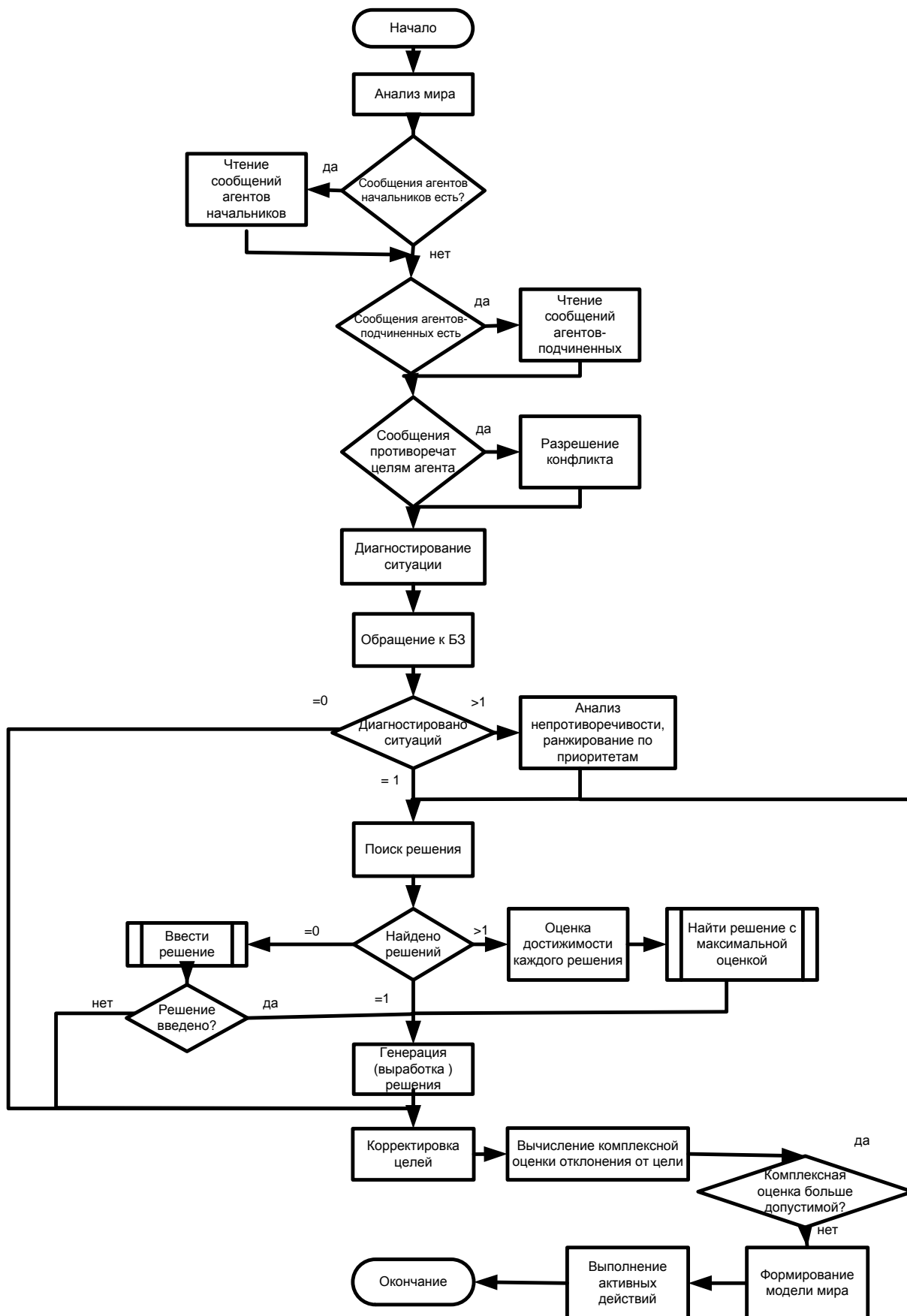


Рис. 12. Блок-схема работы интеллектуального агента

Алгоритм имитатора состоит из следующих основных этапов [36, 55]: определения текущего момента времени  $SysTime = \min T_j, j \in RULE$ ; обработки действий агентов (диагностирования текущих ситуаций, выработки команд управления); формирования очереди правил преобразования; выполнения правил преобразования и изменения состояния рабочей памяти (ресурсов и средств). Для диагностирования текущих ситуаций и выработки команд управления имитатор обращается к модулю экспертной системы.

Более подробней остановимся на данном алгоритме.

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 [25–26] и доказавший свою эффективность и быстроедействие при решении практических задач в системе VPsim. В процессе работы машина вывода *Ips* взаимодействует со следующими очередями [36]:

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  $A_{Agent}(t)$ :

$$Calendar = A_1(t_1), A_2(t_2), \dots, A_j(t_j), \dots, A_m(t_m), \quad (19)$$

$$\forall j: (A_j(t) \in \{A_{Ca}(t)\} \cup \{A_{IN}(t)\} \cup \{A_{OUT}(t)\} \cup \{A_{Agent}\}) \wedge (t_{j-1} \leq t_j \leq t_{j+1}) \quad (20)$$

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

$$Active\_Rules = A_1(t_1), \dots, A_i(t_i), \dots, A_n(t_n), \quad (21)$$

$$\forall i: (t_i = SysTime) \& (A_i(t_i) \in \{A_{OUT}(t)\}), \quad (22)$$

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  $A_{Agent}(t)$ :



$$Wait \& Lock\_Rules = A_1(t_1), \dots, A_k(t_k), \dots, A_z(t_z), \quad (23)$$

$$\forall k : (t_k = SysTime) \& (A_k(t_k) \in \{A_{Ca}(t)\}). \quad (24)$$

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.

$$Agent\_List = AG_1, AG_2, \dots, AG_p, \dots, AG_y, \quad (25)$$

$$\forall p : prior_{p-1} \geq prior_p \geq prior_{p+1}.$$

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;

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

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.

$Execute(Active\_Rules)$  — 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 [36]:

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  $SysTime = SysTime + Dt$ , 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:

$$\forall i(t_{Junction_i}^{out} = SysTime):$$

$$A_{OUT}^{Junction_i} \xrightarrow{ADD} Active\_Rules;$$

2) список завершения активности экземпляров «операция»:

$$\forall k(Status_{Op}^k = "active") \& (t_{Op_k}^{out} = SysTime):$$

$$A_{OUT}^{Op_k} \xrightarrow{ADD} Active\_Rules;$$

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

$$\forall m(t_{receiver_m}^{out} = SysTime): A_{OUT}^{receiver_m} \xrightarrow{ADD} Active\_Rules,$$

$$\forall n(t_{sender_n}^{out} = SysTime): A_{OUT}^{sender_n} \xrightarrow{ADD} Active\_Rules.$$

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 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:

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

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  $Count (Changes\_Queue) \geq 0$ , 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»:

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

2) правила экземпляров «перекресток»:

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

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 выпол-

нении 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. Обработка 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 используем шаг 8.3, а если 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).

**Модель МППР** [48] разработана ее авторами на основе расширения модели процесса преобразования ресурсов (ППР) интеллектуальными агентами и предназначена для моделирования организационно-технических и бизнес-процессов и поддержки принятия управленческих решений. Агент модели МППР име-

ет гибридную архитектуру InteRRap [83], представленную на рис. 13.

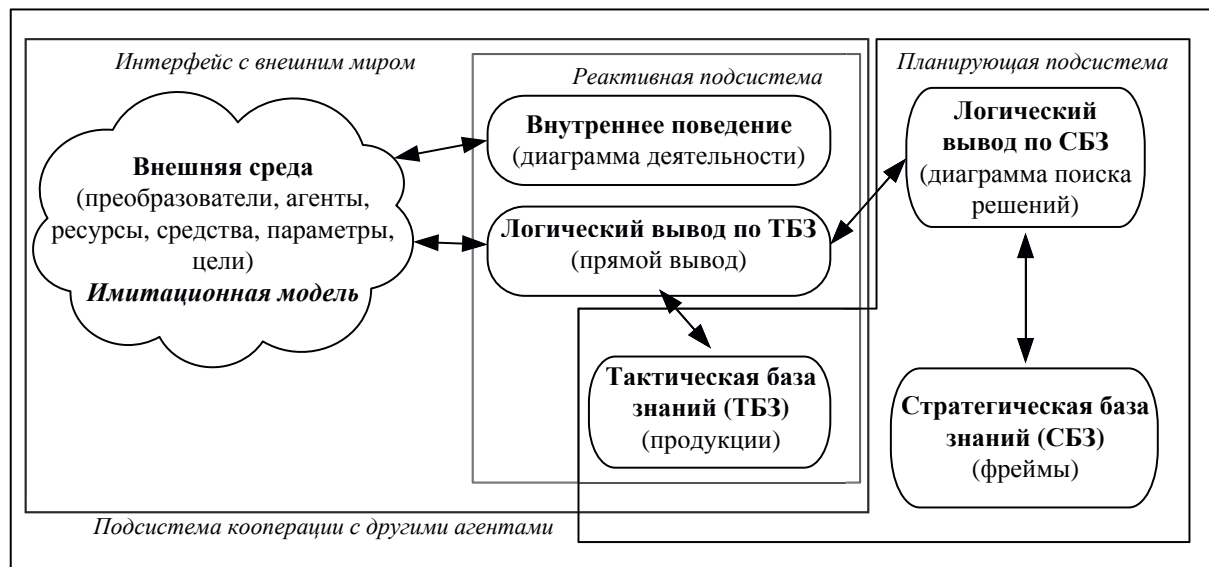


Рис. 13. Гибридная архитектура агента МППР

Классификация агентов процесса преобразования ресурсов

На основе архитектуры гибридного агента МППР можно создавать следующие виды агентов для решения различных прикладных задач [55]:

- реактивные агенты, поведение которых определяется реактивной подсистемой и описывается только диаграммой деятельности (конечным автоматом). Данный вид агентов используется для задач динамического моделирования МППР;

- реактивно-интеллектуальные агенты, поведение которых определяется реактивной подсистемой и описывается только продукционной базой знаний (тактической). Данный вид агентов используется для задач динамического моделирования МППР, описания моделей ЛПП, управляющих процессами;

- интеллектуальные агенты, поведение и логика работы которых определяется только планирующей подсистемой, и знания хранятся в фреймовой базе знаний (стратегической). Данный вид агентов используется для построения проблемно-ориентированных интеллектуальных систем на основе аппарата фреймовых экспертных систем (задачи диагностики, проектирования, построения советующих ЭС и систем технико-экономического проектирования);

- гибридные агенты, полностью реализующие функциональность архитектуры гибридного агента МППР. Данный вид

агентов используется для построения сложных, интеллектуальных систем управления и планирования, контур которых состоит из двух элементов: 1) динамической модели процесса управления и 2) блока (модуля), реализующего поиск решения многопараметрической задачи и выработки (генерации) управляющего воздействия на модель процесса управления.

Модель МППР реализована в СДМС VPsim.MAS [25–26, 36, 55].

Заканчивая обзор и анализ методов моделирования, необходимо отметить то, что задачам технико-экономического проектирования ТЭП ОТС и моделирования бизнеса компании по функциональности и используемым методам наиболее соответствует СДМС VPsim.MAS, реализующая модель МППР, которая обеспечивает выполнение следующих требований:

- описание ситуационной модели в виде дискретной ситуационной сети, как наиболее соответствующей процессам преобразования ресурсов;
- декомпозицию ситуационной модели;
- описание процессов преобразования ресурсов;
- представление информации и моделей с использованием когнитивной графики;
- описание моделей ЛППР в виде интеллектуального агента;
- вывод на знаниях;
- имитационное дискретно-событийное моделирование.

## **6. ВИДЫ СИНТЕЗА ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ И РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ РЕИНЖИНИРИНГА МОДЕЛИ МППР**

Анализ системы в рамках решения задач управления предприятием подразумевает под собой построение аналитиком различных моделей (представлений) системы, отражающих требуемые аспекты ее поведения. Синтез построенных моделей направлен на получение альтернативных вариантов развития системы путем объединения в единый вариант различных подмножеств элементов модели. Будем различать следующие типы синтеза [84]:

- параметрический синтез, при котором исходная модель преобразуется в ряд моделей путем изменения значений отдельных параметров модели;

- структурный синтез, при котором новые модели образуются из исходной путем изменения структуры модели согласно правилам;
- смешанный синтез, подразумевающий выполнение структурного и параметрического синтеза.

Смешанный синтез модели представляет особый интерес, поскольку обеспечивает аналитика методикой прогнозирования развития модели при рассмотрении модели с различных ракурсов, что обеспечивает полноту проводимого синтеза.

Схема проведения смешанного синтеза представлена на рис. 14. Результатом смешанного синтеза является построение дерева моделей, требующих проведения экспериментов и анализа результатов моделирования с целью выявления наиболее эффективного варианта развития системы. Таким образом, полученные в результате смешанного синтеза модели являются планом общего эксперимента по принятию решений в области управления бизнес-системами и организационно-техническими системами (ОТС).

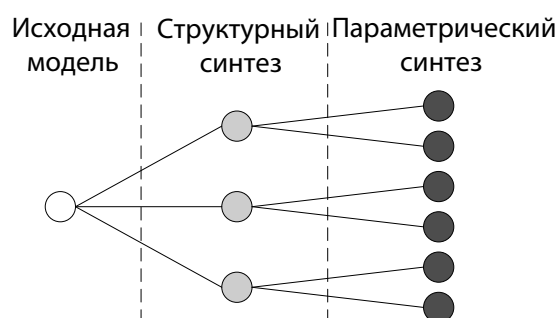


Рис. 14. Схема проведения смешанного синтеза

Концептуальная (содержательная) модель — это абстрактная модель, определяющая структуру моделируемой системы, свойства её элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования. Другими словами, концептуальная модель  $S_k$  — это множество понятий и связей между ними, являющихся смысловой структурой рассматриваемой предметной области:

$$S_k = \langle A, R \rangle, \quad (26)$$

где  $A$  — множество объектов предметной области;

$R$  — множество отношений, связывающих объекты.

Концептуальная модель предметной области визардов анализа и синтеза моделей содержит следующие элементы:



$KMPO_{wizard} = \{Wizard, User, MPPR, ExpertModel, Exp, ExpOut, Analysis, AnalysisOut, Synthesis, SynthesisOut\}$ ,

где

- *Wizard* – множество визардов, разработанных пользователями;
- *User* – множество пользователей, имеющих различные права по созданию, изменению и удалению конкретного визарда;
- *MPPR* – множество элементов модели МППР;
- *ExpertModel* – множество элементов экспертной модели;
- *Exp* – множество экспериментов с моделью МППР;
- *ExpOut* – множество элементов результата эксперимента;
- *Analysis* – множество правил анализа моделей;
- *AnalysisOut* – множество элементов результата анализа;
- *Synthesis* – множество правил синтеза моделей;
- *SynthesisOut* – множество элементов результата синтеза.

Множество элементов модели МППР представляет собой набор:

$MPPR = \{PO, Type, POElem, TaskElem\}$ ,

где

- *PO* – множество предметных областей анализа и синтеза МППР;
- *Type* – используемые справочники модели МППР;
- *POElem* – множество элементов предметных областей (ПО);
- *TaskElem* – множество элементов моделей ПО.

Множество справочников представимо в виде:

$Type = \{TypeRes, TypeMech, Measure\}$ ,

где

- *TypeRes* – множество типов ресурсов;
- *TypeMech* – множество типов средств;
- *Measure* – множество единиц измерений.

Множество элементов ПО представляет собой набор:

$POElem = \{Task, Res, Mech, Order\}$ ,

где

- *Task* – множество моделей ПО;
- *Res* – множество ресурсов ПО;

- *Mech* – множество средств ПО;
- *Order* – множество заявок (сигналов, сообщений) ПО.

Множество элементов моделей ПО имеет следующую структуру:

$$TaskElem = \{TaskGoal, TaskNode, TaskNodeRes, Agent, AgentGoal, AgentSolutionIf, AgentSolutionThen, Tree\_Agent, DataFlow, ScemeItem\},$$

где

- *TaskGoal* – множество целей модели;
- *TaskNode* – множество операций модели;
- *TaskNodeRes* – множество условий запуска, входов и выходов операций;
- *Agent* – множество агентов модели;
- *AgentGoal* – множество целей агентов;
- *AgentSolutionIf* – множество условий «Если» агентов;
- *AgentSolutionThen* – множество условий «То» агентов;
- *Tree\\_Agent* – множество диаграмм деятельности агентов;
- *DataFlow* – множество переходов диаграмм деятельности;
- *ScemeItem* – множество состояний диаграмм деятельности.

Диаграммы деятельности языка UML используются для моделирования реактивного поведения агентов МППР и описывают переходы в рамках выполнения задачи, вызванные внутренними процессами. К основным элементам диаграммы деятельности относят: 1) состояния действия (action state), в которых определены внутренние действия и которые имеют хотя бы один выходящий переход; 2) безусловные переходы, включающие в себя неявное событие – завершение внутреннего действия; 3) условные переходы, включающие помимо завершения действия условия выполнения перехода.

Поведение реактивно-интеллектуальных агентов описывается продукционной базой знаний.

Моделирование интеллектуального поведения агентов МППР осуществляется при помощи продукционных правил «Если-То» и экспертной модели, имеющей следующий вид:

$$ExpertModel = \{UseCaseDiagram, SearchSolutionDiagram\},$$

где

- *UseCaseDiagram* – множество диаграмм прецедентов экспертной модели;

- *SearchSolutionDiagram* – множество диаграмм поиска решений экспертной модели.

Диаграммы поиска решений реализуют вывод на фреймовой экспертной системе. Таким образом, гибридные агенты МППР используют смешанную модель представления знаний.

## 7. ТЕОРЕТИЧЕСКАЯ ОСНОВА МЕТОДА РЕИНЖИНИРИНГА БИЗНЕС-ПРОЦЕССОВ — ОПЕРАЦИОННЫЙ АНАЛИЗ ВЕРОЯТНОСТНЫХ СЕТЕЙ

Применение на практике подходов реинжиниринга бизнес-процессов (РБП) имеет следующие ограничения: 1) требования по высокой квалификации аналитика и его хорошей ориентации в предметной области; 2) отсутствие алгоритмов (правил) проведения РБП. Существующие подходы РБП в первую очередь затрагивают изменения организационной структуры, перераспределения обязанностей, повышения ответственности и мотивации сотрудников, но не дают ответа по влиянию данных изменений на деятельность предприятия и параметры БП, а также не решают задачи анализа узких мест и реинжиниринга процесса; 3) невозможность синтеза решений РБП и проверки выработанных решений на реальном объекте управления; 4) отсутствие автоматизированных средств РБП.

Для решения задачи реинжиниринга динамической модели МППР введены понятия процедур свертки/развертки модели. Под *сверткой* будем понимать процедуру структурного и/или параметрического синтеза модели процессов преобразования ресурсов, которая ведет к уменьшению структуры и/или параметров модели. Процедура *развертки* модели — процедура увеличения размерности модели динамического процесса. Применение данных процедур может привести к улучшению характеристик модели БП.

За основу модели МППР была взята сеть массового обслуживания. Для расчетов сетей массового обслуживания используется теория вероятностных сетей, которая основывается на марковских и полумарковских процессах [86], но большинство результатов получено только для экспоненциальных законов распределения. При количестве узлов сети больше трех для расчетов используются численные приближенные методы. Операционный анализ [87] в отличие от теории массового обслуживания опирается на логику работы рассматриваемой или моде-

лируемой системы. Это позволяет установить простые зависимости между параметрами и показателями работы системы, не абстрагируясь от процессов ее функционирования [85].

**Операционный анализ вероятностных сетей** базируется на следующих принципах [85]:

– все предположения относительно операционных переменных можно проверить измерениями на реальной системе или на ее модели;

– в системе должен существовать баланс потоков: количество требований, которые покинули систему за некоторый период наблюдения, должно равняться количеству требований, которые поступили в систему за этот же период;

– переходы требований от одного узла к другому не должны зависеть от длин очередей в узлах.

**Основные операционные переменные**, которые можно получить или измерениями, или в процессе ИМ системы, следующие [85]:

$q_{0j} (j = \overline{1, K})$  – вероятность (частота) поступления требований в сеть извне к любому узлу ( $K$  – общее количество узлов);

$q_{kj} (j = \overline{1, K})$  – вероятность перехода требований из узла  $k$  к узлу  $j$  ( $k = \overline{1, K}, j = \overline{1, K}$ );

$q_{k0}$  – вероятность того, что после окончания обслуживания в узле  $k$  требований покинут сеть;

$A_k (k = \overline{1, K})$  – количество требований, которые поступили в узел  $k$ ;

$C_{kj} (k = \overline{1, K}, j = \overline{1, K})$  – количество требований, которые покинули узел  $k$  и поступили в узел  $j$ ;

$B_k (k = \overline{1, K})$  – общее время обслуживания требований узлом  $k$ ;  
 $T$  – общее время наблюдения за системой или время моделирования.

Наиболее часто используют следующие **выводимые операционные переменные** [85]:

$$U_k = \frac{B_k}{T}, \quad (27)$$

где  $U_k$  – коэффициент использования узла;

$$S_k = \frac{B_k}{C_k}, \quad (28)$$

где  $S_k$  – среднее время обслуживания в узле  $k$ ;

$$X_k = \frac{C_k}{T}, \quad (29)$$

где  $X_k$  — интенсивность выходящего потока требований из узла  $k$ ;

$$q_{kj} = \begin{cases} \frac{C_{kj}}{C_k}, & k = \overline{1, K} \\ \frac{A_{oj}}{A_j}, & k = 0 \end{cases} \quad \sum_{k=1}^K q_{kj} = 1, \quad (30)$$

где  $q_{kj}$  — относительная частота перехода требований между узлами  $k$  и  $j$ .

Основные результаты операционного анализа формулируются в виде соотношений между операционными переменными. Основой этих соотношений является гипотеза о балансе потоков в сети: *количество требований, которые поступили в некоторый узел на протяжении продолжительного периода  $T$ , равняется количеству требований, которые покинули этот узел.* Гипотеза о балансе позволяет установить зависимости между операционными переменными для каждого узла сети. Эта гипотеза позволяет записать уравнения баланса потоков [85]:

$$X_j = \sum_{k=0}^K X_k q_{kj}, \quad j = \overline{0, K}. \quad (31)$$

Уравнения (31) будут иметь единственное решение для замкнутой сети при заданном  $x_0$ . Для разомкнутой сети уравнения (31) будут линейно зависимыми, однако, и в этом случае они имеют полезную информацию о динамике потоков сети. Производительность узла

$$X_k = \frac{U_k}{S_k}. \quad (32)$$

Коэффициент посещаемости узла  $k$

$$V_k = \frac{X_k}{X_0}. \quad (33)$$

**Среднее время пребывания требования** в вероятностной сети обозначается через  $R$ , а для отдельных узлов — через  $R_k$ . Операционная переменная —  $W_k$  равняется суммарному времени ожидания и времени обслуживания требования узлом  $k$  на протяжении времени  $T$  [85]:

$$R_k = \frac{W_k}{C_k}. \quad (34)$$

Среднее время пребывания в системе выражается через  $R_k$  и коэффициенты посещаемости отдельных узлов [85]:

$$R = \sum_{k=1}^K V_k R_k. \quad (35)$$

В [85] показано, как можно использовать операционный анализ для определения времени пребывания в замкнутой сети (рис. 15). Пусть есть  $M$  устройств, время обслуживания требования любым из них —  $Z$ . Среднее время пребывания требования в сети определяем по формуле

$$R = \frac{M}{x_0} - Z. \quad (36)$$

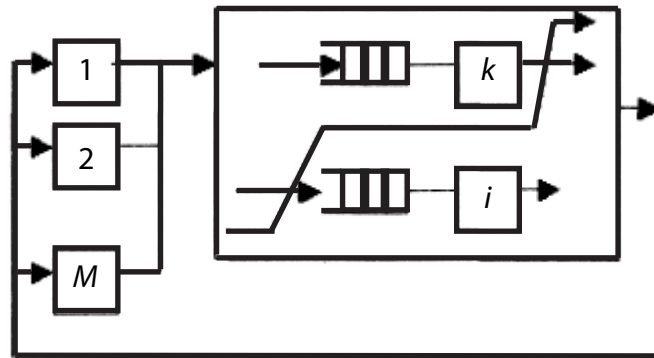


Рис. 15. Определение времени пребывания в замкнутой сети

Среднее время одного цикла взаимодействия, включая время обслуживания требования во внешней сети и пребывание в одном из  $M$  устройств, определяется суммой  $Z + R$ . Общее количество устройств [85]:

$$M = (Z + R)X_0. \quad (37)$$

**Поиск узких мест в сети** является важным аспектом анализа ее работы. Узкое место создается тем узлом сети, у которого коэффициент загрузки  $U$  приближается к единице. В этом узле образуется большая очередь, которая при  $U \geq 1$  становится бесконечной, и сеть переходит в неустойчивый режим работы. Такой узел становится «насыщенным» требованиями. Узкие места в сети обуславливают ее пропускную способность, то есть полностью определяют время пребывания в сети [85].

С точки зрения применения операционного анализа вероятностных сетей к МППР при анализе узких мест необходимо анализировать следующие параметры:

**1. Коэффициент использования узла** (узлам соответствуют *операции* и *агенты*, также необходимо анализировать коэффициент использования *средства*).

**2. Среднюю длительность нахождения заявки в очереди к операции, агенту** (размер очереди заявок к операции  $Q_{Op\_cp}$ , средняя очередь заявок к правилу агента  $Q_{AgR\_cp}$ ).

**3. Коэффициент посещаемости узла и среднюю длительность обработки требования в узле.**

В ходе экспериментальных исследований [88] было установлено следующее: важно анализировать коэффициенты использования не на всем интервале моделирования, а на подынтервалах, соответствующих пикам нагрузки. Также необходимо классифицировать пики нагрузок и для каждого пика искать свои решения.

*Таким образом, операционный анализ вероятностных сетей массового обслуживания можно эффективно применять при анализе узких мест метода реинжиниринга.*

## **8. АЛГОРИТМ ПРОВЕДЕНИЯ РЕИНЖИНИРИНГА МОДЕЛИ МППР (АНАЛИЗА И СТРУКТУРНОГО СИНТЕЗА МОДЕЛИ)**

Рассмотрим исходную модель МППР (бизнес-системы / ОТС)  $M_0$ , предназначенную для анализа процессов, протекающих в системе. В результате проведения эксперимента формируется статистика выполнения операций, функционирования агентов, расходования и формирования ресурсов и заявок, использования средств в операциях.

Для оценки выполнения операции  $Op$  рассмотрим следующие ее параметры: среднюю очередь заявок к операции  $Q_{Op\_cp}$ , среднюю загрузженность операции  $U_{Op\_cp}$ , простой операции из-за отсутствия средств  $P_{MechOp}$ , простой операции из-за отсутствия входных ресурсов  $P_{ResOp}$ :

$$Q_{Op\_cp} = \frac{\sum_{t=1}^{T_{END}} Q_{op}(t)}{T_{END}},$$

$$U_{OP\_cp} = (N \cdot T_{OP}) / T_{END},$$

$$P_{MechOp} = \frac{\sum_{t=1}^{T_{END}} Tact(t) | Count\_Mech\_UnLock(t) < Count\_Mech\_Use}{T_{END}},$$

$$P_{ResOp} = \frac{\sum_{t=1}^{T_{END}} Tact(t) | Count\_Res(t) < Count\_Res\_In}{T_{END}},$$

где  $T_{END}$  – время окончания моделирования,  
 $Round$  – функция взятия целой части вещественного числа,  
 $N$  – количество выполнений операции  $Op$  за время моделирования  $T_{END}$ ,

$T_{OP}$  – длительность выполнения операции  $Op$ ,

$Tact$  – машинный такт имитации модели,

$Count\_Mech\_UnLock$  – количество единиц средства, не заблокированное при выполнении текущих операций,

$Count\_Mech\_Use$  – количество единиц средства, необходимое для запуска операции  $Op$ ,

$Count\_Res$  – текущее количество единиц ресурса,

$Count\_Res\_In$  – количество единиц ресурса, необходимое для запуска операции  $Op$ .

Аналогично оценке очереди проводится оценка среднего состояния ресурсов (как входных, так и выходных по отношению к определенной операции или правилу агента).

Для оценки использования средства в операциях модели рассмотрим среднюю загрузженность средства  $U_{Mech\_cp}$ :

$$U_{Mech\_cp} = \frac{\sum_{t=1}^{T_{END}} Count\_Mech\_Lock(t)}{Count\_Mech \cdot T_{END}},$$

где  $Count\_Mech\_Lock$  – количество единиц средства, заблокированное при выполнении текущих операций,

$Count\_Mech$  – общее количество единиц средства.

Статистику функционирования агента будем анализировать исходя из средней очереди заявок к агенту  $Q_{Ag\_cp}$  и средней загрузженности агента по обработке заявок  $U_{Ag\_cp}$ :



$$Q_{Ag\_cp} = \frac{\sum_{t=1}^{T_{END}} Q_{Ag}(t)}{T_{END}},$$

$$U_{Ag\_cp} = \frac{\sum_{t=1}^{T_{END}} Tact(t) | (\exists \langle MessOp \rangle \in \{AgSolutIf(t) \cup AgSolutThen(t)\}) \& (AgSolutIf(t) = True)}{T_{END}},$$

где  $\langle MessOp \rangle$  – оператор обработки заявки (создание, удаление или блокировка заявки),

$AgSolutIf$  – условия агента «Если»,

$AgSolutThen$  – условия агента «То».

Значения выделенных выходных характеристик модели разбиваются *критическими точками* на интервалы, представленные на рис. 16. Выделим следующие критические точки, определяемые пользователем: точки  $K_{Op1}$ – $K_{Op8}$  для параметров выполнения операции, точки  $K_{Mech1}$  и  $K_{Mech2}$  для параметров использования средства, точки  $K_{Ag1}$ – $K_{Ag3}$  для параметров функционирования агента.

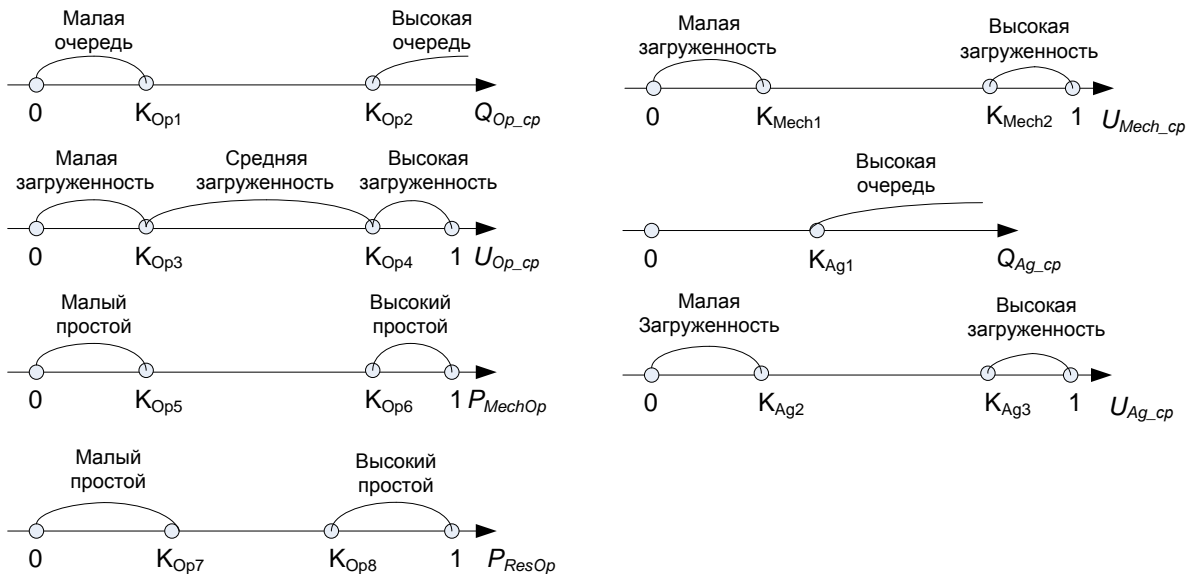


Рис. 16. Интервалы значений выходных характеристик модели

Рассмотрим представление описанного алгоритма проведения анализа и смешанного синтеза в виде графов поиска решений применения операторов синтеза (см. рис. 17). Верши-

ны графа имеют следующие обозначения: 0 – нулевое значение, М – малое значение, С – среднее значение, В – высокое значение соответствующего объекта графа (очереди, загруженности или простоя). Пунктирные линии переходов графа соответствуют решениям для нулевой и малой очередей заявок к операции, сплошные линии – решениям в оставшихся случаях.

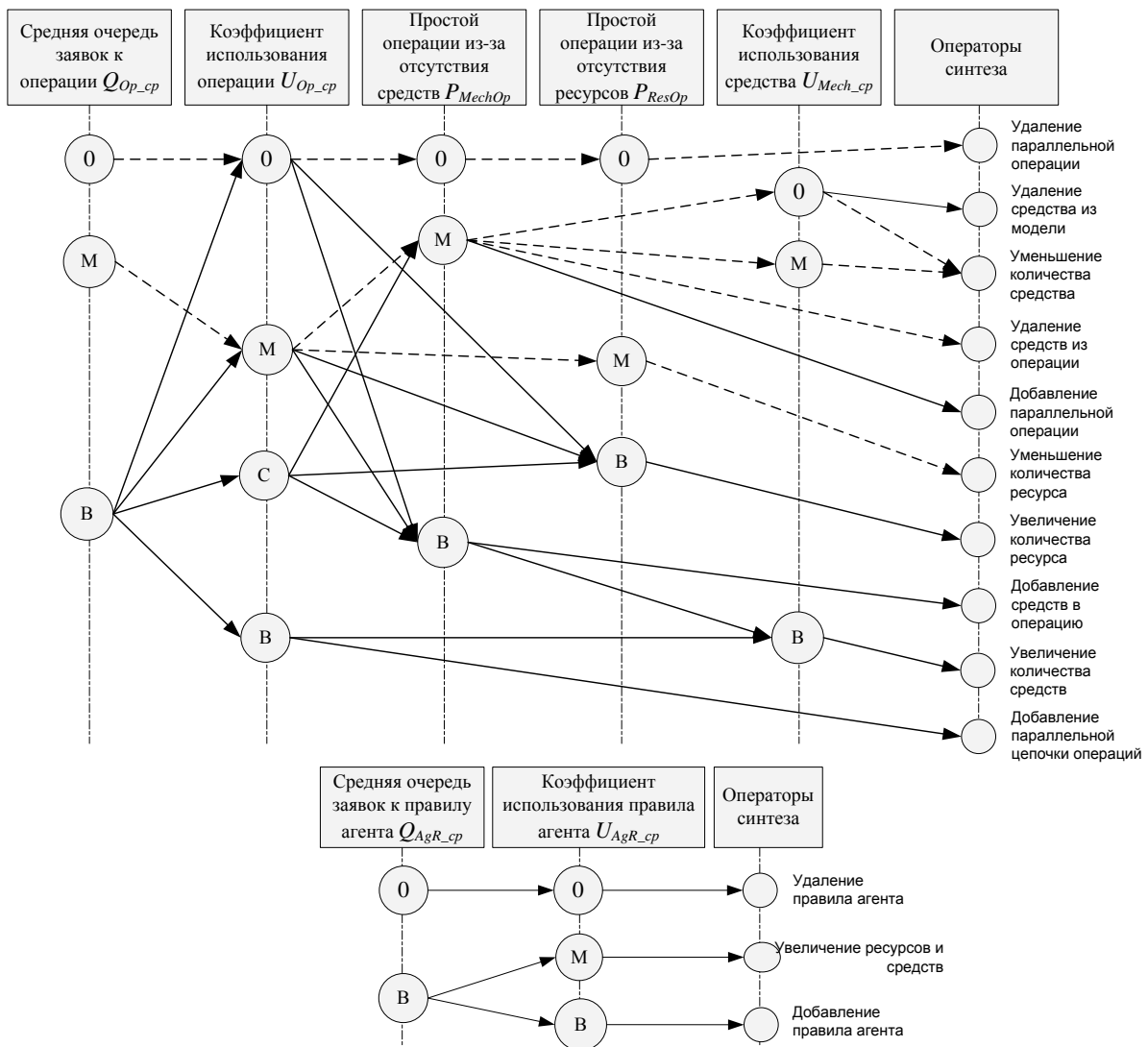


Рис. 17. Графы поиска решений применения операторов синтеза

Результаты проведения синтеза хранятся в классе «Результаты синтеза». На основании данных выделенного класса осуществляется изменение структуры модели  $M_s$  (копии исходной модели  $M_0$ ).

Рассмотрим пример реинжиниринга бизнес-процессов P1..P4, схема выполнения которых представлена слева на рис. 18.

Задачу о выявлении узких мест функционирования такой системы удобно решать при помощи методов имитационного моделирования.

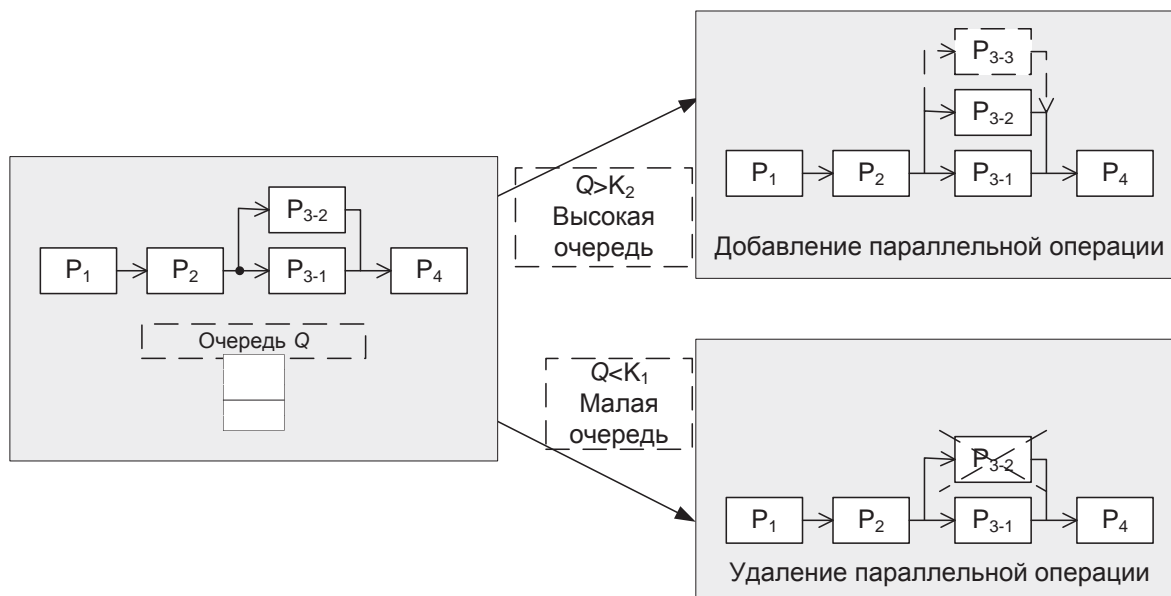


Рис. 18. Схема реинжиниринга узких мест организации бизнес-процессов

Лицо, принимающее решение, в ходе имитации модели анализирует очередь  $Q$  из объектов системы, ожидающих обработки параллельными процессами  $P_{3-1}$  и  $P_{3-2}$ . В случае превышения значения очереди  $Q$  некоего порогового значения  $K_1$  (что означает перегруженность блока  $P_3$ ) принимается решение о добавлении параллельно к существующим нового процесса  $P_{3-3}$ , обеспечивающего приемлемые значения очереди  $Q$  и улучшенные показатели эффективности работы системы (справа вверху на рис. 18). В случае недобора значением очереди  $Q$  порогового значения  $K_2$  (что означает простой блока  $P_3$ ) принимается решение о ликвидации одного из параллельных процессов  $P_{3-2}$  (справа внизу на рис. 18). Анализируя, таким образом, зависимость показателей эффективности работы системы от структуры ее бизнес-процессов, лицо, принимающее решение, находит наиболее эффективную организацию процессов системы.

Применение операционного анализа вероятностных сетей к мультиагентной модели позволило решить задачу уменьшения количества экспериментов, проводимых с моделью

МППР, путем построения модели сети массового обслуживания на основе результатов экспериментов модели МППР, с целью быстрого решения задачи нахождения среднего количества работающих устройств (средств, согласно терминологии МППР).

Разработан алгоритм проведения реинжиниринга модели МППР (анализа и структурного синтеза модели). Исходная модель МППР (БП / ОТС)  $M_0$ , предназначена для анализа процессов, протекающих в системе. В результате проведения эксперимента с моделью МППР формируется статистика выполнения операций, функционирования агентов, расходования и формирования ресурсов и заявок и использования средств в операциях. По результатам анализа экспериментов диагностируются узкие места и принимается решение о свертке/развертке модели. Критерием остановки метода реинжиниринга мультиагентной модели ОТС является снижение времени ожидания до допустимых значений по всем блокам модели.

Элементарный акт реализации метода реинжиниринга будет иметь следующий вид:

$$SM \xrightarrow{Sv(i)} TM,$$

где  $SM$  – исходная модель (*source model*);

$TM$  – новая (измененная) модель (*turned model*);

$Sv(i)$  – элементарное изменение  $Sv(i) \in SV$  как способ воздействия на  $SM$ .

Таким образом, структурное изменение модели осуществляется либо удалением операции, либо добавлением параллельной операции, добавлением средства исполнения операции или ресурсов, добавлением или удалением правила агента, удалением агента. Пользователь может согласиться с предложенным изменением или отказаться от него.

На рис. 19 представлена блок-схема метода принятия решений задачи реинжиниринга модели МППР, которая может служить основой для подготовки соответствующей программной реализации.

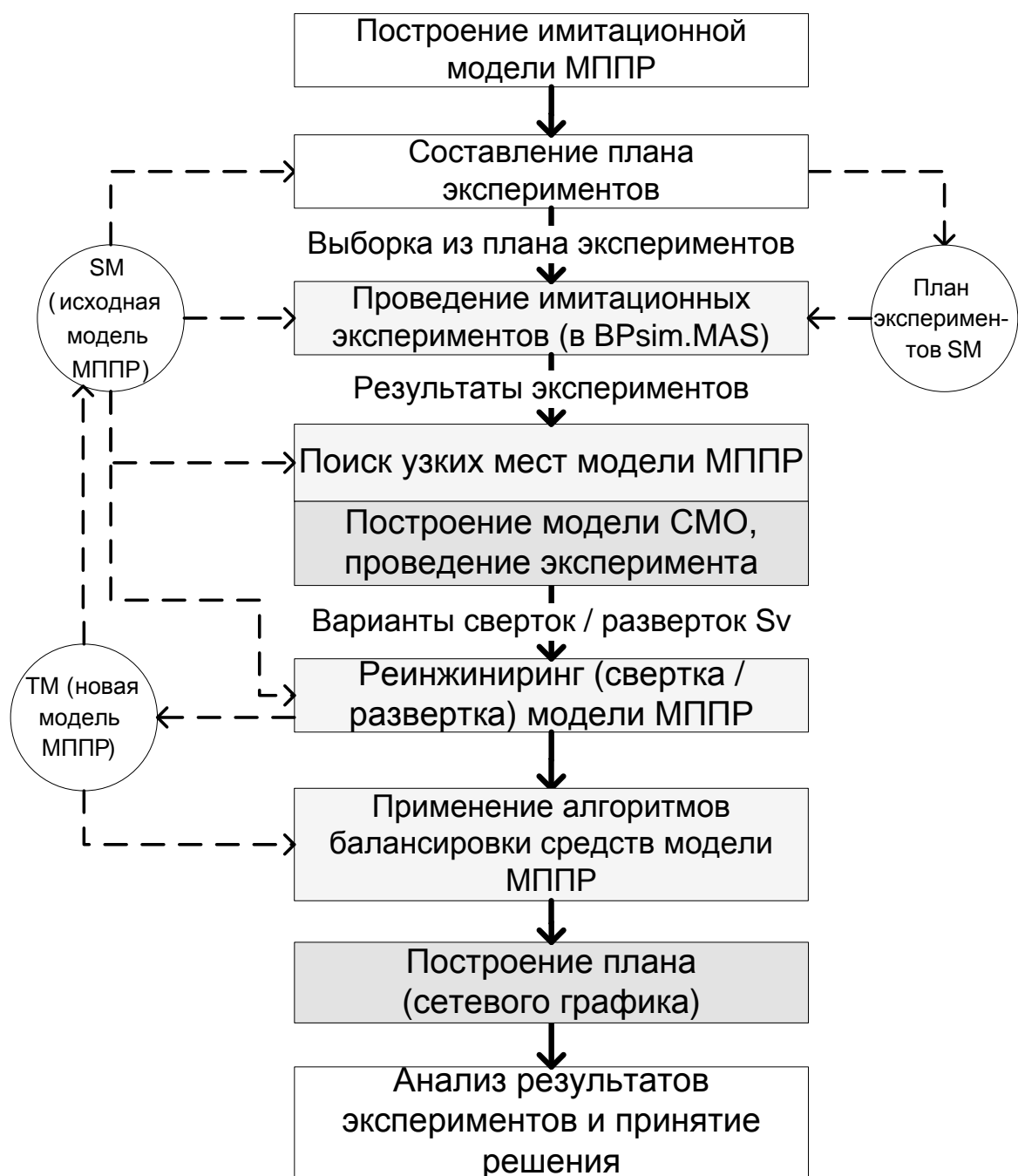


Рис. 19. Общая схема метода принятия решений задачи реинжиниринга модели мультиагентного процесса преобразования ресурсов

Рассмотрим основные этапы метода. Метод проходил отладку при решении задачи моделирования деятельности строительного холдинга, результаты представлены в главе 4. Наиболее эффективно метод позволяет решать задачи планирования для предметных областей строительства, управления проектами и производства на заказ [88]. Предлагаемый метод состоит из следующих этапов:

## 1. Построение имитационной модели (ИМ) МППР

При построении ИМ строятся следующие подмодели:

- генерации объектов (*объектов строительства / проектов / заказов*), такой объект в модели МППР представим в виде экземпляра заявки (транзакта) с набором атрибутов;

- процессов прохождения объектов (*процессов строительства / этапов проекта / производственных операций*), в модели МППР маршрут обработки заявки формируется цепочкой блоков, состоящих из операций и агентов;

- поставок потребляемых ресурсов (*сырья, материалов и полуфабрикатов*), в модели МППР маршрут поставки ресурсов формируется цепочкой блоков, состоящих из операций и агентов;

- работы средств (*станки, оборудование, агрегаты, транспорт, персонал*);

- продаж, в модели МППР маршрут взаимодействия с клиентом формируется цепочкой блоков, состоящих из операций и агентов.

*Подмодель прохождения объектов.* Так, например, при построении подмодели строительства описывается модель типового объекта строительства, которая является параметрически настраиваемой и зависит от типа и свойств сгенерированного объекта (заявки). Так, отдельные блоки модели соответствуют этапам строительства. При поступлении в блок заявки (объекта строительства), в зависимости от ее параметров, происходит вычисление времени длительности этапа, количества потребляемых ресурсов и используемых средств.

Данные предметные области для моделирования набора объектов (*портфеля проектов, возводимых объектов строительства, портфеля производственных заказов*) диктуют следующие специфические требования и соответствующие им правила построения ИМ (проиллюстрируем на примере строительства):

- 1) ограничение по объему субподрядных работ. В случае превышения ограничения на суммарные затраты данный объект строительства становится нерентабельным — проект может быть исключен из портфеля;

- 2) должна использоваться стратегия строительства объектов «первым пришел — первым вышел», т. к. задержки при строи-

тельстве отдельного объекта приводят к ряду дополнительных затрат, связанных с обеспечением проекта (охрана, электроэнергия и т. д.). Завершение процесса строительства объекта и соответственно сокращение его сроков приводят к ускорению возврата инвестиций. В связи с этим блоки модели должны применять выталкивающую стратегию для заявки «объект строительства» (приоритет блоков модели возрастает от начальных этапов строительства к завершающим);

3) для процесса строительства характерно параллельное во времени выполнение этапов разных работ. Работы (соответствующие блоки ИМ), относящиеся к критическому пути, должны иметь приоритет выше, чем у параллельных работ.

Данные требования предметной области и правила построения ИМ также хорошо согласуются с выводами Дэйвиса [89]: «... правило упорядочения, в соответствии с которым первой выполняется работа с наименьшим резервом (или эквивалентное правило минимизации самого позднего времени начала), в среднем дает наилучший результат».

Для моделей БП и ОТС, в которых при недостатке собственных средств возможно привлечение субподрядчиков, разработана подмодель субподряда на основе реактивных интеллектуальных агентов. Если применение субподрядчиков невозможно, то на шаге 7 могут быть использованы эвристические алгоритмы балансировки средств для совершенствования модели МППР.

В подмодели субподряда возможны 2 варианта: 1) чужие средства привлекаются в случае нехватки своих на выполнение всей операции от начала и до окончания; 2) на каждом последующем такте происходит пересмотр возможности отказа от привлечения субподряда. ИМ с агентами субподряда характеризуется следующим: 1) при применении варианта 2 агентов субподряда время ожидания не увеличивается ввиду недостатка средств, т. е. срок проекта по критерию использования средств минимален; 2) по результатам экспериментов можно скорректировать среднее количество своего парка средств и оценить максимальный объем привлечения субподряда; 3) в случае достижения требуемой пропускной способности модели и уменьшения времен ожидания до допустимых значений можно считать план обработки портфеля проектов идеальным (минимальным по срокам). Ограничения реальных проектов строительства: 1) не

всегда допустимо применение субподряда; 2) объем привлечения субподряда имеет ограничения. Применение агентов субподряда позволяет только решить задачу узких мест на средствах мультиагентной модели. Для решения задачи анализа узких мест, возникающих на операциях и ресурсах, необходимо выполнение последующих шагов, завершающихся реинжинирингом мультиагентной модели.

## **2. Составление плана экспериментов**

Планирование эксперимента в соответствии с выдвинутыми гипотезами. При планировании, например, строительных работ могут быть использованы несколько разных моделей субподряда.

## **3. Проведение имитационных экспериментов**

Имитационные эксперименты проводятся в СДМС VPsim.MAS.

## **4. Поиск узких мест модели МППР**

При диагностике узких мест анализируются следующие параметры мультиагентной модели: 1) коэффициент использования операции, средства, агента; 2) среднее время заявки в очереди к операции, агенту; 3) простой операции из-за отсутствия средств и/или входных ресурсов. Для оценки динамики работы операции и агента также анализируется средняя очередь заявок к операции, агенту, а также среднее состояние ресурсов.

На данном этапе осуществляется анализ структуры и параметров исходной модели SM и выявляется допустимость применения правил синтеза к тем или иным блокам исходной модели SM или ее параметрам.

## **5. Построение модели сети массового обслуживания (СМО), проведение эксперимента**

Применение операционного анализа вероятностных сетей к модели МППР позволяет также решить задачу уменьшения количества экспериментов путем построения модели СМО на основе результатов экспериментов модели МППР, с целью быстрого нахождения среднего количества работающих устройств (средств МППР).

## **6. Реинжиниринг (свертка / развертка) модели МППР**

По результатам анализа статистики экспериментов диагностируются узкие места и принимается решение о свертке/



развертке модели. Критерием остановки реинжиниринга модели МППР является снижение времени ожидания до допустимых значений по всем блокам модели. Данный этап направлен на решение задачи распараллеливания сетевых графиков параллельно возводимых во времени объектов строительства или проектов (в блоках ИМ могут возникать ситуации с параллельной обработкой заявок).

### **7. Применение алгоритмов балансировки средств модели МППР**

Если для задачи недопустимо привлечение субподрядчиков по отдельным видам средств, то для равномерности загрузки этих средств могут быть применены эвристические алгоритмы балансировки. Эксперименты проводятся до нахождения эффективного решения.

### **8. Построение плана ( сетевого графика)**

По результатам имитационного эксперимента формируется сетевой график портфеля объектов. Критический путь для каждого отдельного объекта (например, объекта строительства) определяется как выбор из параллельных цепочек работ цепи, не имеющей запаса времени по результатам имитационного эксперимента. Оценка резерва времени для каждой работы выполняется автоматически. Так, для параллельных цепочек (не относящихся к критическому пути) это будет резерв времени последней работы в цепи.

### **9. Анализ результатов экспериментов и принятие решения**

Данный этап зависит от предметной области и решаемой задачи. Так, например, для строительства может быть выполнена проверка соответствия бюджету портфеля проектов и срокам. Если сроки и бюджет портфеля проектов (портфель объектов строительства) удовлетворяют, то завершаем проведение экспериментов, иначе возвращаемся на шаг 7 или корректируем условия задачи и переходим на шаг 1.

*Проверка адекватности новой модели.* Проверка адекватности осуществляется на тестовых экспериментах путем сравнения результатов расчета новой модели  $TM$  с результатами эксперимента на изучаемом объекте и исходной модели  $SM$  (при одинаковых условиях). Это позволяет установить границы применимости новой модели  $TM$ .

## 9. ОБЗОР СИСТЕМ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ СИТУАЦИЙ (СДМС)

### 9.1. Обзор существующих СДМС

Ниже приводится обзор близких по функциональности систем.

*ARIS ToolSet* – система проектирования и моделирования бизнес-процессов. Обзор сделан по материалам [4; 18].

Система ARIS представляет собой комплекс средств анализа и моделирования деятельности предприятия, а также разработки автоматизированных информационных систем. ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

- организационные: представляют структуру системы – иерархию организационных подразделений, должностей и конкретных лиц, многообразие связей между ними;
- функциональные: содержат иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
- информационные: отражают структуру информации, необходимую для реализации всей совокупности функций системы;
- модели управления, представляющие комплексный взгляд на реализацию процесса.

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 (OMT) и т. п.

Для описания бизнес-процессов в ARIS используется общая ARIS-модель процесса [18] (см. рис. 20).

В дополнении к общей ARIS-модели для описания процессов используется стандарт EPC (extended Event Driven Process Chain) – «расширенная нотация описания цепочки процесса», управляемого событиями. В рамках нотации используются следующие объекты:

- функция — служит для описания операций (работ);
- событие — служит для описания реальных состояний системы, влияющих и управляющих выполнением функций;
- организационная единица — отдел предприятия;
- документ — объект, отражающий реальные носители информации;
- прикладная система — реальная система, используемая в рамках технологии выполнения функций;
- стрелка связи между объектами — объект описывает тип отношений между другими объектами.

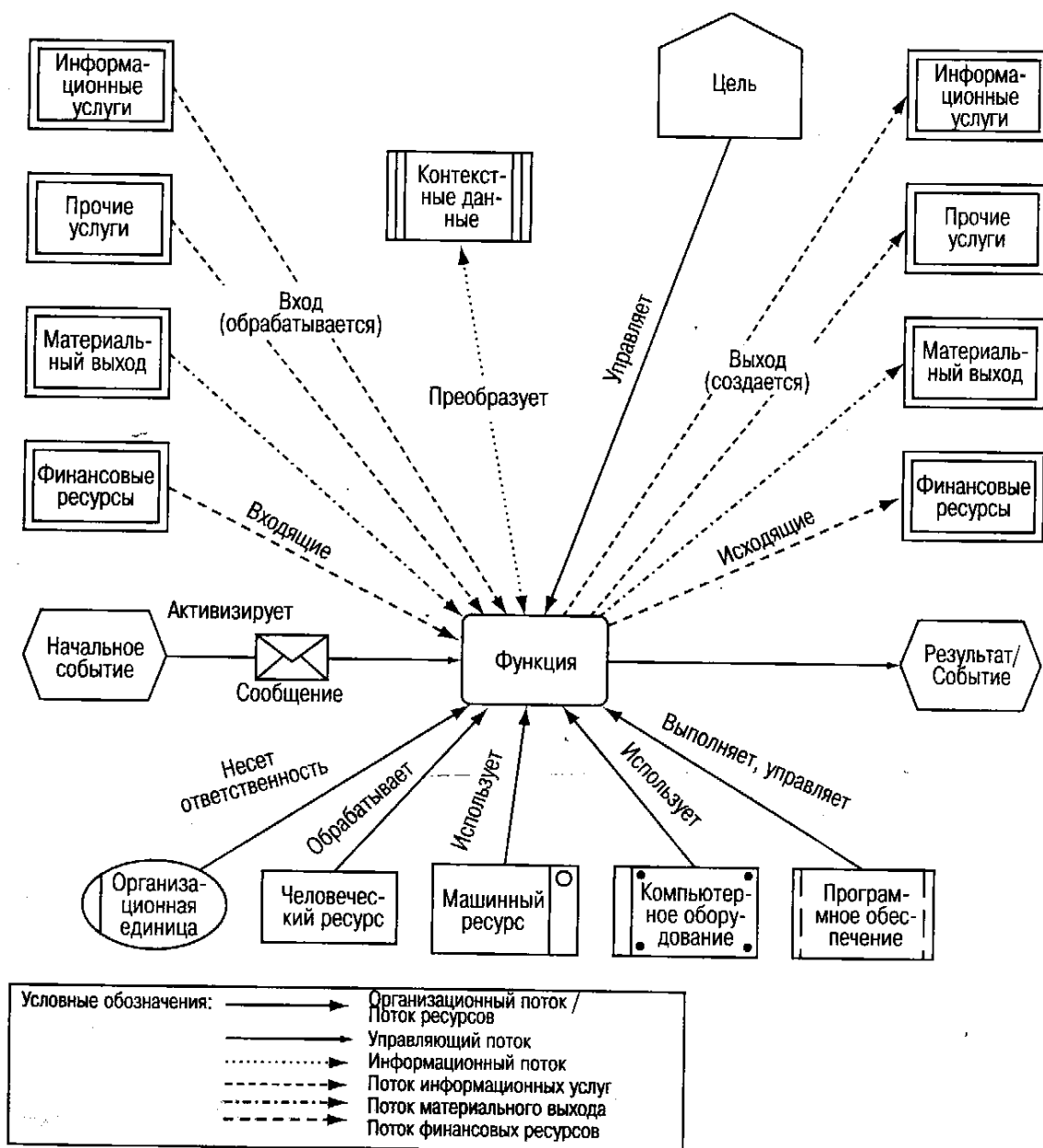


Рис. 20. Общая ARIS-модель процесса

Помимо указанных выше основных объектов, при построении диаграммы ЕРС могут быть использованы многие другие объекты. Применение большого числа различных объектов, объединенных различными типами связей, значительно увеличивает размер модели и делает ее плохо читаемой. Нотация ЕРС построена на определенных семантических правилах описания:

1. Каждая функция должна быть инициирована событием и должна завершаться событием.

2. В каждую функцию не может входить более одной стрелки, «запускающей» выполнение функции, и выходить не более одной стрелки, описывающей завершение выполнения функции.

Процесс в нотации ЕРС представляет собой последовательность процедур, расположенных в порядке их выполнения. Используемые при построении модели символы логики позволяют отразить ветвление и слияние процесса.

По полноте описания бизнес-процесса и разделения на подклассы составляющих процесса данная нотация самая полная. Основные отличия от стандартов IDEF: связь функции с целью, описание начального события, инициирующего процесс, и конечного, сигнализирующего о выполнении процесса.

*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. G2 совмещает выполнение правил и процедур в текущий момент времени со способностями рассуждений через некоторое время.

*Пакет 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. ReThink является проблемно-ориентированным приложением комплекса G2, которое позволяет разработчикам использовать не толь-

ко специализированные средства моделирования процессов, но и универсальные средства комплекса по созданию интеллектуальных объектно-ориентированных систем реального времени [10; 61].

Для представления моделей процессов преобразования ресурсов используются диаграммы, состоящие из блоков и соединений. Блоки представляют операции в процессе преобразования, а соединения — потоки сущностей (например, документы, запасные части, упаковки с отпускаемой продукцией). В системе реализованы стандартные блоки, которые могут быть использованы в качестве сборочных элементов для построения работающей модели практически любого процесса преобразования. Свойства и поведение блоков могут описываться как точными, так и случайными величинами. В случае необходимости разработчик может переопределять поведение блоков или даже задавать новые классы с помощью базовых средств комплекса G2.

ReThink поддерживает создание иерархических моделей, позволяющих описывать процессы с различной степенью детализации. Все элементы моделей, включая ресурсы процесса преобразования, могут модифицироваться непосредственно во время исполнения. Результаты изменений можно увидеть сразу же после их введения.

ReThink позволяет формировать стоимостные и временные характеристики различных проектов для объективного их сравнения, а также проверять гипотезы «что если». Для анализа работы моделей предусмотрен целый набор инструментов: блоки-датчики для сбора данных, блоки-установщики значений атрибутов сущностей, графики для наглядного отображения результатов моделирования. С помощью датчиков можно снимать такие показатели, как длительность цикла обработки сущности на том или ином этапе, стоимость обработки, а также любые другие свойства, определенные разработчиком.

Для проверки гипотез «что если» в системе реализован механизм сценариев. Сценарии позволяют исследовать зависимость поведения одной и той же модели от поведения внешнего мира (частоты поступления заявок, их сложности и т. д.) и каких-либо параметров этой модели (например, количества транспортных средств или численности служащих, занятых оформлением заказов). Варьируемые параметры и измеряемые показатели вы-

носятся на отдельное окно сценария, после чего в результате прогона модели автоматически формируется отчет. Кроме этого, ReThink позволяет использовать сценарии для объективного сравнения альтернативных проектов: один и тот же сценарий описывает некоторое заранее заданное поведение внешнего мира и используется для проведения экспериментов различных моделей. Результаты экспериментов, внесенные в отчет, являются основой для сопоставления этих моделей.

#### *Система имитационного моделирования AnyLogic*

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. Основным элементом модели в среде AnyLogic является активный объект. Он имеет внутреннюю структуру и поведение, а также может содержать другие активные объекты как свои элементы. Структура активного объекта зависит от того, из каких элементов он состоит и какие связи существуют между включенными в него объектами. Поведение объекта показывает его реакцию на внешние события в виде последовательности его изменений во времени. AnyLogic позволяет [1; 6; 57–59]:

- моделировать при помощи визуальных, гибких, расширяемых, повторно-используемых объектов (стандартных и своих), а также на языке Java;

- увеличить жизненный цикл модели, быстро подстраивая её к меняющимся условиям, при решении которых необходимы как высокий, так и низкий уровни абстракции;

- использовать мощный арсенал средств анализа и оптимизации непосредственно из среды разработки модели;

- интегрировать модель открытой архитектуры с офисными и корпоративными ПО (электронные таблицы, БД, ERP и CRM системы);

- эффектно представлять свои результаты, сопровождая модель интерактивной анимацией, а также давая возможность доступа к модели через Интернет.

AnyLogic поддерживает на единой платформе следующие существующие подходы: дискретно-событийного и непрерывного моделирования (блок-схемы процессов, системную динамику, агентное моделирование, карты состояний, системы уравнений).

AnyLogic имеет развитый базовый язык дискретного и смешанного дискретно/непрерывного моделирования, на основе которого построены решения для конкретных областей; библиотека Enterprise Library, а также Material Flow Library (потoki материалов) и Healthcare Library (работа медицинских учреждений), включённые в состав продукта. Enterprise Library содержит традиционные объекты: очереди, задержки, конвейеры, ресурсы и т. п., так что модель и анимация строятся в стиле drag-and-drop и гибко параметризуются.

Агенты AnyLogic могут создаваться и уничтожаться динамически, перемещаться, общаться друг с другом. При помощи агентов моделируют рынки (агент — потенциальный покупатель), конкуренцию, цепочки поставок (агент — компания), население (агент — семья, житель города или избиратель). Только агентные модели позволяют получить представление об общем поведении системы, исходя из предположений о поведении ее элементов при отсутствии знания о глобальных законах, т. е. в наиболее общем случае.

AnyLogic предлагает множество средств для описания структуры, поведения и данных моделируемой системы: объекты, интерфейсы и иерархия, блочные диаграммы, карты состояний, таймеры, порты и передача сообщений, переменные и алгебраические дифференциальные уравнения, а также возможность добавить выражение, оператор или функцию на языке Java в любом месте модели.

Поскольку модели AnyLogic — 100% Java, их можно не только запускать на многих платформах, но и помещать на сайты в виде апплетов. Это свойство позволяет удаленным пользователям запускать интерактивные модели в web-браузере.

В AnyLogic представление модели является визуальным и иерархическим. Простой графический язык моделирования, основанный на UML-RT, оперирует понятиями объектов и связей между ними — дискретными (отправка сообщений произвольной структуры) и непрерывными (отслеживание показателей). Для описания сложного поведения пользователь может использовать графические диаграммы переходов и состояний. Такие диаграммы позволяют визуально проектировать сложные бизнес-процессы и многошаговые действия с альтернативами.

Описание поведения объектов производится с помощью фрагментов кода на языке Java: пользователю необходимо определить существенный код действий в специальных полях свойств элементов объектов, а весь рутинный код генерируется пакетом автоматически. Когда базовых возможностей AnyLogic недостаточно, разработчик модели может использовать Java для создания дополнительных классов. При разработке сложных моделей не удастся 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 динамического моделирования ситуаций VPsim.MAS*

VPsim.MAS [36; 55; 62] – проблемно-ориентированная СДМС, позволяющая адекватно описывать и моделировать экономические, производственные, технические, информационные и бизнес-процессы в рамках мультиагентных процессов преобразования ресурсов. СДМС VPsim.MAS обеспечивает выполнение следующих функций:

- 1) проектирование концептуальной модели предметной области;
- 2) создание динамической модели мультиагентного процесса преобразования ресурсов;
- 3) поддержка методики ССП;
- 4) динамическое моделирование;
- 5) анализ результатов экспериментов (получение отчетов по моделям и результатам экспериментов, экспорт результатов экспериментов в MS Excel и MS Project).

В состав СДМС VPsim.MAS входит универсальная оболочка ЭС «Конструктор фрейм-систем», преимуществом которой 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 БЗ реализуются на языке Transact-SQL. Таким образом, не потребовалось создание языка вывода на фреймовой модели. Данный фактор снижает требования к навыкам системных программистов, аналитиков и инженеров по знаниям, поддерживающих работоспособность данной системы, а также позволяет повысить эффективность их работы [36]. В целом СДМС VPsim.MAS (за исключением оболочки ЭС) является про-



блемно-ориентированной, за счет чего для работы с нею не требуются навыки программирования.

## 9.2. Требования к СДМС

На основе предыдущих разделов выделим следующие требования к средствам СДМС:

1. Проектирование концептуальной модели предметной области.
2. Описание динамических процессов преобразования ресурсов:
  - a) Описание ресурсов, средств, преобразователей.
  - b) Описание целей
    - i. В виде графа.
    - ii. 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 мультиагентных моделей. Наличие агентов (моделей ЛПР), обладающих моделью поведения и знаниями:
  - a) Наличие класса (элемента) «агент», на основе которого можно создавать интеллектуального агента.
  - b) Модель поведения агента (язык описания сценариев поведения агентов, язык описания команд управления).
  - c) Язык описания знаний агента.
  - d) Язык описания целей агента.
  - e) Язык обмена сообщениями между агентами.
6. Поддержка имитационного моделирования. Интегрируемость СДМС со средством имитационного моделирования.
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.
9. Стоимость.

### 9.3. Сравнительный анализ СДМС

Результаты сравнения функциональных возможностей пакетов представлены в таблице.

#### Сравнительный анализ систем, близких по функциональности к СДМС

№	Параметр	ARIS	G2	AnyLogic	BPsim
1	Проектирование концептуальной модели предметной области	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	+
2.3	- Иерархическая модель процесса	+	+	+	+
3	Наличие языка описания команд	НЕТ	+	НЕТ	+
4	Описание модели на ограниченном естественном языке	НЕТ	+	НЕТ	+
5	Построение мультиагентной модели				
5.1	- элемент АГЕНТ	НЕТ	НЕТ	+	+
5.2	- модели поведения агентов	НЕТ	НЕТ	+	+
5.3	- база знаний агента	НЕТ	НЕТ	НЕТ	+
5.4	- язык обмена сообщениями	НЕТ	НЕТ	НЕТ	+
6	Имитационное моделирование	+	+	+	+
7	Экспертное моделирование	НЕТ	+	НЕТ	+
8	Ситуационный подход	НЕТ	+	НЕТ	+
9	Стоимость, тыс. долларов	50	70	4,8	3

Как следует из таблицы и проведенного сравнительного анализа, система BPsim (BPsim.MAS & BPsim.MSS) 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 BPsim. Специализированными средствами поддержки методики BSC обладают только системы ARIS и BPsim, 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 BPsim. С точки зрения пользователя, не обладающего навыками программирования, удобными средствами описания/создания модели мультиагентного процесса преобразования обладает только проблемно-ориентированная СДМС BPsim. 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 языка высокого уровня, благодаря чему пакеты могут предоставлять разработчику моделей серьезный уровень функциональности.

## **ЗАКЛЮЧЕНИЕ**

Рассмотрено моделирование мультиагентных процессов преобразования ресурсов (МППР) в организационно-технических системах. Проведен анализ существующих систем динамического моделирования ситуаций (СДМС). На основе интеграции методов имитационного, экспертного, мультиагентного и ситуационного моделирования разработана модель МППР, которая легла в основу пакетов программ СДМС VPsim.MAS и системы технико-экономического проектирования VPsim.MSS. Возможности пакетов проиллюстрированы на примерах решения задач.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

BSC	Balanced ScoreCard – система сбалансированных показателей
АП	Абсолютный приоритет
БВСЦ	Блок выбора сценария
БД	База данных
БЗ	База знаний
ЗА	Значение атрибута
ИМ	Имитационное моделирование (имитационная модель)
ИНС	Интеллектуальная система
ИО	Информационный объект
ИП	Информация о применении
ИФ	Имя фрейма
КГ	Концептуальный граф
КМПО	Концептуальная модель предметной области
КО	Концептуальное отношение
ЛПР	Лицо, принимающее решения
ЛТП	Логико-трансформационные правила (корреляционные правила)
МАС	Мультиагентная система
МО	Множество определения
МППР	Мультиагентный процесс преобразования ресурсов
МСС	Мультисервисная сеть связи

ОМИА	Основа модели интеллектуального агента
ООП	Объектно-ориентированный подход
ОТС	Организационно-техническая система
ПВ-сети	Сети потребностей и возможностей
ППР	Процесс принятия решений
ПрО	Предметная область
РБП	Реинжиниринг бизнес-процессов
РП	Рабочая память
СА	Структура атрибутов
СК	Структура концептов
СИМ	Система имитационного моделирования
СДМС	Система динамического моделирования ситуаций
СМ	Ситуационная модель
СППР	Система поддержки принятия решений
ССЛ	Структура слотов
ССМ	Система ситуационного моделирования
ССП	Структура сценариев поведения
ТФ	Тип фрейма
ТЭП	Технико-экономическое проектирование
ФК	Фрейм-концепт
ХД	Хранилище данных
ЭС	Экспертная система
ЯПЗ	Язык представления знаний

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Борщев А. В. Практическое агентное моделирование и его место в арсенале аналитика/ А. В. Борщев // Exponenta Pro. 2004. № 3–4.
2. Технология системного моделирования / Е. Ф. Аврамчук [и др.]; под общ. ред. С. В. Емельянова. М.: Машиностроение; Берлин: Техник, 1988. 520 с.
3. 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.
4. Системная интеграция в управленческой деятельности: сборник статей / под ред. С. Л. Гольдштейна. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2002. 309 с.
5. Представление и использование знаний: пер. с япон./ под ред. Х. Уэно, М. Исидзука. М.: Мир, 1989. 220 с.
6. Карпов Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю. Г. Карпов. СПб.: БХВ-Петербург, 2005. 400 с.: ил.
7. Клыков Ю. И. Ситуационное управление большими системами / Ю. И. Клыков. М.: Энергия, 1974. 136 с.
8. Клыков Ю. И. Семиотические основы ситуационного управления / Ю. И. Клыков. М.: МИФИ, 1974. 220 с.
9. Клыков Ю. И. Банки данных для принятия решений / Ю. И. Клыков, Л. Н. Горьков. М.: Сов. радио, 1980. 155 с.
10. 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.

11. Поспелов Д.А. Мышление и автоматы / Д.А. Поспелов, В.Н. Пушкин. М.: Советское радио, 1972. 224 с.
12. Поспелов Д.А. Ситуационное управление: теория и практика / Д.А. Поспелов. М.: Наука, 1986. 288 с.
13. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II: пер. с англ. / А. Прицкер. М.: Мир, 1987. 646 с.
14. Советов Б.Я. Моделирование систем: учеб. для вузов. 3-е изд. / Б.Я. Советов, С.А. Яковлев. М.: Высш.шк., 2001. 343 с.
15. Форрестер Дж. Мировая динамика: пер. с англ. / Дж. Форрестер; под ред. Д.М. Гвишиани, Н.Н. Моисеева. М.: Наука, 1978. 168 с.
16. Филиппович А. Ю. Интеграция ситуационного, имитационного и экспертного моделирования в полиграфии / А. Ю. Филиппович. М., 2003. 310 с.
17. 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.
18. Шеер А.В. Моделирование бизнес-процессов / А.В. Шеер. М.: Весть-Метатехнология, 2000. 205 с.
19. Wooldridge M. Intelligent Agent: Theory and Practice / M. Wooldridge, N. Jennings // Knowledge Engineering Review. 1995. № 10 (2).
20. Greenwald A. Guest Editors' Introduction: Agents and Markets / A. Greenwald, N. Jennings, P. Stone // Intelligent Systems. 2003. Vol.18. p. 12–14.
21. Dash R. Computational-Mechanism Design: A Call to Arms / R. Dash, D. Parkes, N. Jennings // Intelligent Systems. 2003. Vol.18. p. 40–47.
22. Minsky M. A framework for Representing Knowledge in The Psychology of Computer Vision / M. Minsky, McGraw-Hill 1975.
23. Model Checking Rational Agents / R. Bordini, M. Fisher, W. Visser, M. Wooldridge // Intelligent Systems. — 2003. Vol.18. p. 40–47.
24. Злобин Э.В. Управление качеством в образовательной организации / Э.В. Злобин, С.В. Мищенко, Б.И. Герасимов. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2004. 88 с.



25. 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.
26. Аксенов К. А. Принципы построения системы имитационного моделирования процессов преобразования ресурсов VPsim / 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.
27. 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.
28. 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.
29. Аксенов К. А. Решение задачи планирования портфеля проектов и анализа узких мест бизнес-процесса на основе мультиагентного моделирования и метода критического пути / К. А. Аксенов, К. В. Ван, О. П. Аксенова // Современные проблемы науки и образования. – 2014. – № 2; URL: [www.science-education.ru/116-12630](http://www.science-education.ru/116-12630) (дата обращения 25.06.2014).
30. 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.
31. Калянов Г. Н. CASE структурный системный анализ (автоматизация и применение) / Г. Н. Калянов. М.: Лори, 1996. 242 с.
32. Марка Д. SADT – Методология структурного анализа и проектирования: пер. с англ. / Д. Марка, К. МакГоуэн. М.: Метатехнология, 1993. 465 с.

33. Трахтенгерц Э.А. Компьютерные системы поддержки принятия управленческих решений / Э.А. Трахтенгерц // Сборник статей II международной конференции по проблемам управления. М., 2003.
34. Трахтенгерц Э.А. Компьютерная поддержка принятия решений / Э.А. Трахтенгерц. М.: СИНТЕГ, 1998.
35. Мильнер Б.З. Теория организаций / Б.З. Мильнер. М.: ИНФРА-М, 1998. 336 с.
36. Аксенов К.А. Динамическое моделирование мульти-агентных процессов преобразования ресурсов: монография / К.А. Аксенов, Н.В. Гончарова. Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2006. 311 с.
37. Ольве Н.-Г. Оценка эффективности деятельности компании: практическое руководство по использованию сбалансированной системы показателей / Н.-Г. Ольве, Ж. Рой, М. Веттер. М.: Вильямс, 2004. 304 с.
38. Частиков А.П. Разработка экспертных систем. Среда CLIPS / А.П. Частиков, Т.А. Гаврилова, Д.Л. Белов. – СПб.: БХВ-Петербург, 2003. – 608 с.
39. Смирнова Г.Н. Проектирование экономических информационных систем: учебник / Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов. М.: Финансы и статистика, 2001. 512 с.
40. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник. 2-е изд. /А.М. Вендров. М.: Финансы и статистика, 2005. 544 с.
41. Маклаков С.В. VPwin и Erwin. CASE – средства разработки информационных систем / С.В. Маклаков. М.: ДИАЛОГ-МИФИ, 1999. 256 с.
42. Описание CASE-средств. Режим доступа: [www.interface.ru](http://www.interface.ru).
43. UML – The Unified Modeling Language. Режим доступа: <http://www.uml.org>.
44. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон. – М.: ДМК, 1993.
45. 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.
46. Шенк Р. Обработка концептуальной информации: пер. с англ. / Р. Шенк. М.: Энергия, 1980. 360 с.

47. 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.
48. Джексон Питер. Введение в экспертные системы: уч. пос. пер. с англ. / Питер Джексон. М.: Издательский дом «Вильямс», 2001. 624 с.
49. Sowa J.F. Conceptual graphs for a database interface / J.F. Sowa // IBM Journal of Research and Development. 1976. № 20. P. 336–357.
50. Sowa J.F. Conceptual Structures: Information Processing in Mind and Machine / J.F. Sowa. – Reading, MA: Addison – Wesley, 1984. 481 p.
51. Sowa J.F. Knowledge Representation: Logical, Philosophical, and Computational Foundations / J.F. Sowa. – Pacific Grove, CA: Brooks/ Cole Publishing Co., 2000. 594 p.
52. 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.
53. Руководство пользователя «Система технико-экономического проектирования мультисервисных сетей связи (СТЭП МСС) «VPsim.MSN». Екатеринбург, 2008. – 44 с.
54. Кельтон В. Имитационное моделирование. Классика CS. 3-е изд. / В. Кельтон, А. Лоу. СПб.: Питер; Киев: Издательская группа BHV, 2004. 847 с.
55. Аксенов К. А. Теория и практика средств поддержки принятия решений: монография / К. А. Аксенов. Germany, Saarbrücken: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2011. 341 с.
56. Гнеденко Б. Д. Введение в теорию массового обслуживания / Б. Д. Гнеденко, И. Н. Коваленко. М.: Наука, 1987. 336 с.
57. AnyLogic. Учебное пособие по агентному моделированию. Режим доступа: [www.xjtek.com](http://www.xjtek.com).
58. Borshchev A.V. Java Engine for UML Based Hybrid State Machines / A.V. Borshchev, Y.B. Kolesov, Y.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>.
59. Borshchev A. AnyLogic 4.0: Simulating Hybrid Systems with Extended UML-RT/ A. Borshchev // Simulation News Europe, No. 31 April 2001, pp 15–16 Режим доступа: <http://www.xjtek.com/files/papers/hybridumlsne2001.pdf>.
60. Маклаков С. Имитационное моделирование с Arena / С. Маклаков. Компьютер-пресс, 2001. № 7. С. 135–136.
61. Описание системы G2. Режим доступа: [www.gensym.com](http://www.gensym.com).
62. Руководство пользователя «Руководство пользователя системы динамического моделирования ситуаций VPsim.MAS». Екатеринбург, 2008. 82 с.
63. Гаврилова Т. А. Состояние и перспективы разработки баз знаний интеллектуальных систем / Т. А. Гаврилова // Журнал «Новости искусственного интеллекта». М., 1996. № 1. С. 5–43.
64. Кузнецов И. П. Кибернетические диалоговые системы / И. П. Кузнецов. М.: Наука, 1976. 293 с.
65. Искусственный интеллект. Кн. 2: Модели и методы / под ред. Д. А. Поспелова. М.: Радио и связь, 1990.
66. Андрейчиков А. В. Интеллектуальные информационные системы: учебник / А. В. Андрейчиков, О. Н. Андрейчикова. М.: Финансы и статистика, 2004. 424 с.
67. Поспелов Д. А. Многоагентные системы – настоящее и будущее / Д. А. Поспелов // Информационные технологии и вычислительные системы. 1998. № 1.
68. Энциклопедический словарь Ф. А. Брокгауза и И. А. Ефрона. Репринтное изд. СПб.: Полрадис, 1993. 480 с.
69. Jennings N. R. On agent-based software engineering / N. R. Jennings // Artificial Intelligence. 2000, vol. 117. P. 277–296. URL: <http://www.agentfactory.com/~rem/day4/Papers/AOSE-Jennings.pdf> (дата обращения: 02.06.2013).
70. Wooldridge M. Agent-based software engineering / M. Wooldridge // IEEE Proc. Software Engineering. 1997. № 144 (1) P. 26–37.
71. Мультиагентное моделирование и планирование логистики / К. А. Аксенов, А. Л. Неволина, О. П. Аксенова, Е. Ф. Смолий // Современные проблемы науки и образования. 2013. № 4; URL: [www.science-education.ru/110-9744](http://www.science-education.ru/110-9744) (дата обращения: 16.08.2013).

72. Скобелев П. О. Мультиагентные технологии для управления ресурсами в реальном времени / П. О. Скобелев // Механика, управление и информатика (Таруса, 2–4 марта 2011 г.) — Таруса, 2011; URL: [http://www.iki.rssi.ru/seminar/2011030204/presentation/20110303\\_03.pdf](http://www.iki.rssi.ru/seminar/2011030204/presentation/20110303_03.pdf) (дата обращения: 02.06.2013).
73. Rzevski G. MAGENTA Technology: A Family of Multi-Agent Intelligent Schedulers / G. Rzevski, J. Himoff, P. Skobelev // International conference on multi-agent systems // Proceedings of Workshop on Software Agents in Information Systems and Industrial Applications 2 (SAISIA). Fraunhofer IITB, Germany, February 2006. URL: <http://rzevski.net/06%20i-Scheduler%20Family.pdf> (дата обращения: 02.06.2013).
74. Vittikh V. A. Multiagent Interaction Models for Constructing the Needs-and-Means Networks in Open Systems. Automation and Remote Control / V. A. Vittikh, P. O. Skobelev. Vol. 64, 2003, pp. 162–169. <http://link.springer.com/article/10.1023%2FA%3A1021836811441> (accessed 02 June 2013).
75. Скобелев П. О. Открытые мультиагентные системы для поддержки процессов принятия решений при управлении предприятиями / П. О. Скобелев // Известия Самарского научного центра Российской академии наук. 2001. Т. 3, № 1. С. 71–79.
76. Инструментальные средства для открытых сетей агентов / В. И. Городецкий, О. В. Карсаев, В. В. Самойлов, С. В. Серебряков // Известия РАН. «Теория и Системы Управления». М.: Наука, 2008. № . 3. С. 106–124.
77. Городецкий В. И. Управление нагрузкой грид на основе многоагентной самоорганизации. Ч.1: Мехатроника, Автоматизация, Управление / В. И. Городецкий, О. Л. Бухвалов. 2011. № 3. С. 40–46.
78. Городецкий В. И., Бухвалов О. Л. Управление нагрузкой грид на основе многоагентной самоорганизации. Ч.2: Мехатроника, Автоматизация, Управление / В. И. Городецкий, О. Л. Бухвалов. 2011. № 7. С. 20–25.
79. Городецкий В. И. Открытые многоагентные системы и самоорганизация: Новые возможности / В. И. Городецкий // Тринадцатая национальная конференция по искусственному интеллекту КИИ-2012, Белгород, 18 октября <http://www.raai.org/resurs/papers/kii-2012/present/Gorodetsky.pdf>.

80. Городецкий В.И. Средства спецификации и инструментальной поддержки командного поведения автономных агентов / В.И. Городецкий, С.В. Серебряков, Д.В. Троцкий // Известия Южного федерального университета. 2011. № 3. С. 116–133.
81. Multi-Agent Technology for Air Traffic Control and Incident Management in Airport Airspace. Proceedings of the International Workshop «Agents in Traffic and Transportation» / V. Gorodetskiy, O. Karsaev, V. Samoilov, V. Skormin. Estoril, Portugal, IEEE Computer Press. 2008. pp. 119–125.
82. Gorodetsky V. Agent Mining: The Synergy of Agent and Data Mining / V. Gorodetsky, L. Cao, P. Mitkas // International Journal «IEEE Intelligent Systems», May/June 2009, pp. 64–72.
83. Jorg P. Muller «The Agent Architecture InteRRap: Concept and Application» / Jorg P. Muller, Markus Pischel. German Research Center for Artificial Intelligence (DFKI)
84. Аксенов К.А. Анализ и синтез процессов преобразования ресурсов на основе имитационного моделирования и интеллектуальных агентов / К.А. Аксенов, А.С. Антонова, И.А. Спицина // Научно-технические ведомости СПбГПУ № 1 (115): Информатика. Телекоммуникации. Управление. СПб., 2011. С. 13–20.
85. Томашевский В. Имитационное моделирование в среде GPSS / В. Томашевский, Е. Жданова. — М.: Бестселлер, 2003. — 416 с.
86. Клейнрок Л. Теория массового обслуживания / Л. Клейнрок. — М.: Машиностроение, 1979. — 432 с.
87. Литвин В.Г. Анализ производительности мультипрограммных ЭВМ / В.Г. Литвин, В.П. Аладышев, А.И. Винниченко. — М.: Финансы и статистика, 1984. — 159 с.
88. Использование аппарата операционного анализа вероятностных сетей для определения среднего количества приборов обслуживания мультиагентной модели / Кай Ван, К.А. Аксенов, О.П. Аксенова, М.В. Киселёва // Современные проблемы науки и образования. — 2012. — № 3; URL: <http://www.science-education.ru/103-6290> (дата обращения: 22.05.2012).
89. Исследование операций / под ред. Дж. Моудера, С. Элмграби. — М.: Мир, 1981. Т. 2. — 677 с.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ОБЗОР МЕТОДОВ И СРЕДСТВ СИТУАЦИОННОГО МОДЕЛИРОВАНИЯ МУЛЬТИАГЕНТНЫХ ПРОЦЕССОВ ПРЕОБРАЗОВАНИЯ РЕСУРСОВ (МППР) .....	6
1. СИТУАЦИОННОЕ УПРАВЛЕНИЕ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИМИ СИСТЕМАМИ (ОТС).....	6
1.1. Ситуационный подход в управлении .....	6
1.2. Рассмотрение организационно-технических систем с точки зрения процессов преобразования ресурсов.....	8
2. ПРОЦЕСС ПРИНЯТИЯ РЕШЕНИЙ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ УПРАВЛЕНИЯ.....	14
2.1. Системы поддержки принятия решений (СППР).....	17
2.2. СППР в стратегическом управлении .....	19
3. СИСТЕМНЫЙ АНАЛИЗ ОТС.....	21
4. АНАЛИЗ И ВЫБОР МОДЕЛЕЙ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ .....	25
4.1. Анализ фреймовых моделей.....	26
4.2. Фреймовый подход Швецова для построения концептуальной модели предметной области (КМПО).....	28
4.3. Применение диаграмм последовательности для визуализации вывода на сети фрейм-концептов и концептуальных графах (диаграммы поиска решений) .....	30
5. МЕТОДЫ МОДЕЛИРОВАНИЯ .....	32
5.1. Имитационное моделирование .....	33
5.2. Экспертное моделирование .....	35
5.3. Ситуационное моделирование .....	37
5.4. Мультиагентный подход.....	39

5.5. Модель мультиагентных процессов преобразования ресурсов (МППР).....	45
6. ВИДЫ СИНТЕЗА ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ И РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ РЕИНЖИНИРИНГА МОДЕЛИ МППР .....	62
7. ТЕОРЕТИЧЕСКАЯ ОСНОВА МЕТОДА РЕИНЖИНИРИНГА БИЗНЕС-ПРОЦЕССОВ — ОПЕРАЦИОННЫЙ АНАЛИЗ ВЕРОЯТНОСТНЫХ СЕТЕЙ.....	66
8. АЛГОРИТМ ПРОВЕДЕНИЯ РЕИНЖИНИРИНГА МОДЕЛИ МППР (АНАЛИЗА И СТРУКТУРНОГО СИНТЕЗА МОДЕЛИ) .....	70
9. ОБЗОР СИСТЕМ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ СИТУАЦИЙ (СДМС).....	81
9.1. Обзор существующих СДМС .....	81
9.2. Требования к СДМС .....	88
9.3. Сравнительный анализ СДМС.....	89
ЗАКЛЮЧЕНИЕ .....	91
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ.....	92
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	94



*Учебное издание*

**Аксенов Константин Александрович,  
Гончарова Наталья Вадимовна**

## **МОДЕЛИРОВАНИЕ И ПРИНЯТИЕ РЕШЕНИЙ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ**

**Часть 1**

Редактор *Н. П. Кубыщенко*  
Компьютерный набор *К. А. Аксенова*  
Компьютерная верстка *Е. В. Суховой*

Подписано в печать 28.01.2015. Формат 60×84 1/16.  
Бумага типографская. Плоская печать. Усл. печ. л. 6,0.  
Уч.-изд. л. 6,5. Тираж 50 экз. Заказ № 11.

Издательство Уральского университета  
Редакционно-издательский отдел ИПЦ УрФУ  
620049, Екатеринбург, ул. С. Ковалевской, 5  
Тел.: 8 (343) 375-48-25, 375-46-85, 374-19-41  
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ  
620075, Екатеринбург, ул. Тургенева, 4  
Тел.: 8 (343) 350-56-64, 350-90-13  
Факс: 8 (343) 358-93-06  
E-mail: press-urfu@mail.ru

