

**С.А.Юдицкий, П.Н.Владиславлев**

---

# **ОСНОВЫ предпроектного анализа организационных систем**

Рекомендовано  
Учебно-методическим объединением  
по образованию в области прикладной информатики  
в качестве учебного пособия  
для студентов высших учебных заведений,  
обучающихся по специальности 351400  
“Прикладная информатика (по областям)”



**МОСКВА  
“ФИНАНСЫ И СТАТИСТИКА”  
2005**

УДК 005.521:334.7

ББК 65.29№6

Ю16

**РЕЦЕНЗЕНТЫ:**

**Кафедра проектирования экономических  
информационных систем  
Московского государственного университета  
экономики, статистики и информатики  
(заведующий кафедрой – Ю.Ф. Тельнов,  
доктор экономических наук, профессор);  
Ю.Л. Барон,  
кандидат технических наук, научный консультант  
Министерства промышленности и энергетики РФ**

**Юдицкий С.А., Владиславлев П.Н.**  
Ю16 **Основы предпроектного анализа организационных систем:**  
Учеб. пособие. – М.: Финансы и статистика, 2005. – 144 с.  
ISBN 5-279-02930-0

Описывается комплекс подготовительных работ, предшествующих проектированию: определяются цели, поставленные перед системой, а также способы (сценарии) достижения целей и планирование реализации проекта. Рассматриваются последовательные этапы предпроектного анализа организационных систем: целеполагание, факторное, операционное и потоковое моделирование.

Для студентов и аспирантов вузов, обучающихся по специальности «Прикладная информатика (по областям)», а также для разработчиков сложных проектов, преподавателей, научных работников и других специалистов, интересующихся проблематикой системного моделирования.

Ю  $\frac{240400000 - 069}{010(01) - 2005}$  334 - 2004

УДК 005.521:334.7  
ББК 65.29№6

ISBN 5-279-02930-0

© Юдицкий С.А., Владиславлев П.Н., 2005

# ОГЛАВЛЕНИЕ

---

Предисловие .....	5
Введение .....	7
<b>Глава 1. Целеполагание при предпроектном анализе организационных систем .....</b>	<b>14</b>
1.1. Общая характеристика технологии целеполагания .....	14
1.2. Структурирование целей системы .....	18
1.3. «Взвешивание» целей системы .....	19
1.4. Минимизация числа локальных целей системы .....	23
1.5. Классификация многоцелевых альтернатив .....	27
1.6. Определение в классе оптимальной альтернативы ....	30
<b>Глава 2. Факторное моделирование организационных систем .....</b>	<b>35</b>
2.1. Принципы факторного моделирования .....	35
2.2. Выбор управляющих факторов .....	40
2.3. Построение когнитивной карты и формирование начальных условий .....	45
2.4. Прогнозирование развития системы при различных стратегиях и выбор стратегии .....	46
<b>Глава 3. Операционное моделирование организационных систем .....</b>	<b>49</b>
3.1. Операционный принцип моделирования функционирования систем .....	49
3.2. Сети Петри – формальный аппарат операционного моделирования .....	53
3.2.1. Анализ сетей Петри на основе ленты достижимости .....	54
3.2.2. Редукция сети Петри .....	59
3.3. Моделирование целевого сценария системы на основе графа операций .....	63
3.3.1. Описание целевого сценария в виде сети Петри ....	63
3.3.2. Формальное описание событий, инициирующих переходы на графе операций .....	67

3.3.3. Равносильные преобразования описания событий .....	70
3.3.4. Моделирование функционирования процессов при «нештатных» ситуациях .....	73
3.3.5. Имитационное моделирование целевого сценария на основе графа операций .....	75
3.4. Моделирование сценария логического управления распределенных систем на основе взаимодействующих графов операций .....	80
3.4.1. Временные взаимодействия графов операций .....	80
3.4.2. Обменные взаимодействия графов операций .....	82
<b>Г л а в а 4. Потокосное моделирование организационных систем .....</b>	<b>90</b>
4.1. Принципы потокосного моделирования .....	90
4.2. Описание потокосного сценария организационных систем .....	96
4.3. Пассивное потокосное моделирование организационных систем .....	102
4.4. Активное потокосное моделирование организационных систем .....	114
4.4.1. Описание жизненного цикла объектов для классов эволюционно-потокосной диаграммы .....	119
4.4.2. Описание внутрикласовых и межекласовых преобразований в активной потокосной модели ...	122
4.4.3. Оценка достижимости целей по результатам моделирования эволюционно-потокосной диаграммы .....	127
4.4.4. Механизм функционирования динамической модели, реализующей эволюционно-потокосную диаграмму .....	130
4.4.5. Представление результатов активного потокосного моделирования в виде конвейерно-временной диаграммы .....	133
<b>Приложение. О проблемах логического моделирования и путях их решения .....</b>	<b>138</b>
<b>Литература .....</b>	<b>142</b>

# **ПРЕДИСЛОВИЕ**

---

Одной из современных тенденций в области разработки и реализации проектов создания и развития организационных систем является возрастание значимости стадии предпроектного анализа и моделирования. Это связано, во-первых, с увеличением сложности систем и соответственно цены ошибок, допускаемых на различных стадиях реализации проектов (в первую очередь ошибок концептуального, стратегического характера). Во-вторых, целенаправленность и стратегическая выверенность проектов становятся одними из важнейших факторов, обеспечивающих конкурентные преимущества создаваемых систем в условиях динамичной внешней среды. Не случайно в последние годы и в сфере бизнеса, и в сфере государственного управления повышается востребованность соответствующих разработок, в частности, с использованием сценарных моделей, различных дорожных карт стратегического развития («roadmaps»), а также методов стратегического планирования с помощью карт сбалансированных показателей и т.д.

На стадии предпроектного анализа формируются основные ориентиры будущего проекта: принимаются рамочные решения по структуре и принципам функционирования организационной системы, формируются требования к управлению, обеспеченности ресурсами, взаимодействию с внешней средой и т.д. Все эти задачи эффективно решаются с помощью информационных технологий, основанных на компьютерном моделировании. Работы в области предпроектного анализа и моделирования в течение ряда лет ведутся в Институте проблем управления им. В.А. Трапезникова РАН (совместно с другими организациями).

Цель данного учебного пособия, предназначенного для студентов и аспирантов вузов по специальности «Прикладная информатика (по областям)», – систематизация обобщенных теоретически и апробированных на практике подходов, моделей и методов предпроектного анализа организационных систем.

При написании учебного пособия авторы старались в доступной и лаконичной форме, с иллюстрацией на конкретных примерах изложить целостную сквозную методологию предпроектного анализа организационных систем, дающую ответы на такие, в частности, вопросы: Какие цели определяют функционирование системы? (Под *целями*, как это принято, понимают показатели, на основе которых оценивают результаты функционирования системы.) Какие возможные стратегии могут быть использованы для достижения поставленных целей? Какие сценарии могут быть реализованы при функционировании системы? Какой может быть динамика потоков (финансовых, информационных, товарно-сырьевых, энергетических и т.д.) при функционировании системы? и др.

Пособие состоит из четырех глав, посвященных соответственно:

- *целеполаганию* – определению наиболее значимых целей организационной системы и уровней их достижимости исходя из реальных ограничений на ресурсы;

- *факторному моделированию* – выбору факторов, определяющих развитие организационной системы, прогнозированию развития при различных стратегиях и выбору стратегии;

- *операционному моделированию* – формальному описанию сценария достижения выбранных целей при выбранной стратегии, проведению аналитического и имитационного исследования сценария;

- *потокосому моделированию* – построению и анализу формальных моделей, отображающих динамику потоков в организационной системе.

Каждая глава книги завершается списком контрольных вопросов и заданий, позволяющих читателю, во-первых, проверить, адекватно ли он понял изложенный материал и, во-вторых, может ли он творчески применить полученные знания для решения новых задач.

Авторы выражают признательность заведующему кафедрой проектирования экономических информационных систем Московского государственного университета экономики, статистики и информатики доктору экономических наук, профессору Ю.Ф. Тельнову и научному консультанту Министерства промышленности и энергетики РФ кандидату технических наук Ю.Л. Барону, которые высказали ряд полезных замечаний.

## **ВВЕДЕНИЕ**

---

Фундаментальными характеристиками организационной системы являются ее структура (архитектура) и способ функционирования (поведение). При создании новых систем и реформировании/модернизации существующих надо прежде всего установить цели, поставленные перед системой, понять, достижимы ли они при реальных ограничениях на финансовые и временные ресурсы, и если достижимы, то какими должны быть изменения во времени в процессе достижения поставленных целей, т.е. целевая динамика системы.

При создании (реформировании, модернизации) сложных организационных систем приходится сталкиваться с множеством серьезных (и часто неожиданных) проблем и решать эти проблемы с учетом сложившейся ситуации уже в ходе выполнения проекта. Это проявляется в дополнительных финансовых и ресурсных затратах, в корректировке целей проекта, характеристик проектируемой системы, в сдвиге сроков и т.д. Хорошо известно также, что чем на более позднем этапе проектирования обнаружена несогласованность или ошибка, тем больше потери и выше цена их устранения. Добавим к этому, что пока способность «хорошо» управлять проектом, своевременно выявляя и устраняя, а еще лучше не допуская ошибок, является в большей степени искусством, чем наукой, и определяется опытом и талантом руководителей и исполнителей проекта.

Вместе с тем в современной теории управления и информатике получен ряд интересных результатов (подходов, моделей, методов), адаптация и интеграция которых позволяют, на наш взгляд, ввести в рассмотрение в качестве начальной фазы создания (реформирования/модернизации) сложных организационных систем так называемое предпроектное моделирование (предпроектный анализ). Предпроектное моделирование предшествует принятым фазам жизненного цикла проекта (концепция, разработка, реализация, завершение) и должно оперировать строгим формальным аппаратом для выбора целей, анализа целевой и потоковой динамики, прогнозирования тенденций и сценариев развития системы. Желательно, чтобы аппарат предпроектного моделирования мог быть применен как для анализа функционирования конкретных систем,

являющихся результатом проектирования, так и для моделирования развития проектов, которые интерпретируются как производящие системы (т.е. системы, производящие другие системы).

Опора предпроектного моделирования на строгий формальный аппарат является предпосылкой для создания практически удобного инструментария и информационных технологий аналитического и имитационного компьютерного моделирования на начальной фазе построения системы (хотя мы и отдаем себе отчет в том, что создание рабочей версии такого инструментария потребует серьезных затрат).

Предлагаемое читателю учебное пособие преследует цель – дать общее системное представление о предпроектном моделировании (с акцентом на динамическое моделирование и примененный для этого формальный аппарат), а также более подробно описать результаты в этой области, полученные непосредственно авторами.

В общем случае предпроектное моделирование функционирования любой системы состоит из ряда последовательных этапов, где каждый последующий этап развивает и расширяет результаты, полученные на предыдущем этапе.

**Э т а п 1. Целеполагание**, под которым понимается выбор целей, поставленных перед системой\*. В рамках целеполагания осуществляются поиск и изучение информации о системе и среде, в которой система функционирует, формируются основные (глобальные) цели и проводится их структурирование с выделением элементарных (локальных) целей. На основе экспертных оценок определяются рейтинги (сравнительная значимость) локальных целей и степень (уровень) их достижимости, зависящая от ограничений на потребляемые ресурсы. Формальными методами осуществляется минимизация числа локальных целей исходя из заданной степени достижимости глобальной цели, формируется множество возможных альтернатив целевых решений на базе отобранных локальных целей с учетом уровня их достижимости, и из их числа выбирается оптимальная альтернатива.

**Э т а п 2. Факторное моделирование** позволяет, основываясь на целевых факторах (и уровнях их достижимости), определенных на этапе целеполагания, выбрать стратегию развития организаци-

---

\* Существуют различные толкования термина «целеполагание». В частности, расширенное толкование, подразумевающее широкий спектр работ в области предпроектного анализа. В данной книге целеполагание трактуется в более узком смысле – буквально как выбор целей.

онной системы на заданном временном интервале. Для этого с помощью экспертов устанавливается множество управляющих факторов, влияющих на достижимость отобранных целевых факторов, выделяются наиболее весомые управляющие факторы, и для каждой возможной стратегии моделируется изменение на временном интервале выбранных целевых и управляющих факторов. Отсюда и название этапа – «факторное моделирование». Выбирается та стратегия, которая порождает состояние системы в конечной точке интервала моделирования с наилучшим значением специального показателя – индекса результативности.

Факторное моделирование дает предварительный (грубый) экспертный прогноз развития системы.

*Э т а п 3. Операционное моделирование* системы, развивая результаты факторного моделирования, исследует процессы, направленные на достижение выбранных целей в соответствии с выбранной стратегией, происходящие внутри системы и на ее стыке с внешней средой, на более детальном (структурном) уровне. Термин «операционное моделирование» отражает тот факт, что процессы, протекающие в системах, состоят из отдельных действий – операций, выполняемых в определенном порядке. Результатом этого этапа предпроектного моделирования является операционный сценарий системы, базирующийся на наборе операций и их характеристиках, на формальном описании событий, инициирующих переходы.

Операционный сценарий описывается на основе формально-го аппарата сетей Петри [6,10] и графов операций [20].

Различают два вида операционных сценариев: целевой, отображающий динамику достижения целей при выполнении операций (вклад каждой операции), и сценарий логического управления, задающий алгоритм функционирования системы с учетом ее архитектуры (состава структурных компонентов и связей между ними). Сценарий логического управления отображает как порядок следования операций, так и взаимодействия (временные и обменные) между процессами, реализуемыми в компонентах системы.

При операционном моделировании выявляются аномалии в алгоритме функционирования и архитектуре организационной системы и осуществляется необходимая коррекция.

*Э т а п 4. Завершающий этап предпроектного моделирования – потоковое моделирование* – связан с исследованием динамики финансовых, информационных, материальных и иных потоков,

циркулирующих в системе и между системой и внешней средой. Его результатом является так называемый потоковый сценарий системы.

Потоки соответствуют как мобильным объектам, пропускаемым через систему (заказы, изделия и их компоненты, перемещаемые грузы), так и ресурсам, обеспечивающим жизнедеятельность процессов (сырье, энергия, человеческий фактор). Потоковый сценарий отображает взаимодействие операций, введенных на этапе операционного моделирования, с накопителями элементов (порций) потоков.

Накопители подразделяются на пассивные, для которых моделируются только внесение и изъятие элементов потока, и на активные, для которых, помимо внесения/изъятия, моделируется и жизненный цикл элементов внутри накопителя.

Потоковое моделирование на базе пассивных и активных накопителей опирается на разный математический аппарат (соответственно сетей Петри и эволюционно-потоковых диаграмм [16]) и образует два направления.

В рамках потокового моделирования:

- составляется формальное описание потокового сценария системы на основе графических средств, развивающих нотации технологии структурного анализа и проектирования SADT [9];
- на основе потокового сценария строится пассивная либо активная потоковая модель системы;
- проводится имитационное моделирование потоковых моделей и фиксируются его результаты (в виде конвейерно-временных диаграмм [16]).

Схема предпроектного моделирования организационных систем представлена на рисунке, где прямоугольники со сплошным контуром обозначают последовательные этапы предпроектного моделирования: целеполагание, факторное, операционное и потоковое моделирование; ромбики – принятие решения о способе продолжения моделирования:

- повторение данного этапа с уточненными данными, сформированными экспертами, либо возврат к какому-нибудь из предыдущих этапов (исходящая из ромбика горизонтальная стрелка);
- переход к следующему этапу моделирования организационной системы (исходящая вертикальная стрелка).

Собственно процесс проектирования организационной системы и его связи с этапами предпроектного моделирования на рисунке изображены пунктиром.

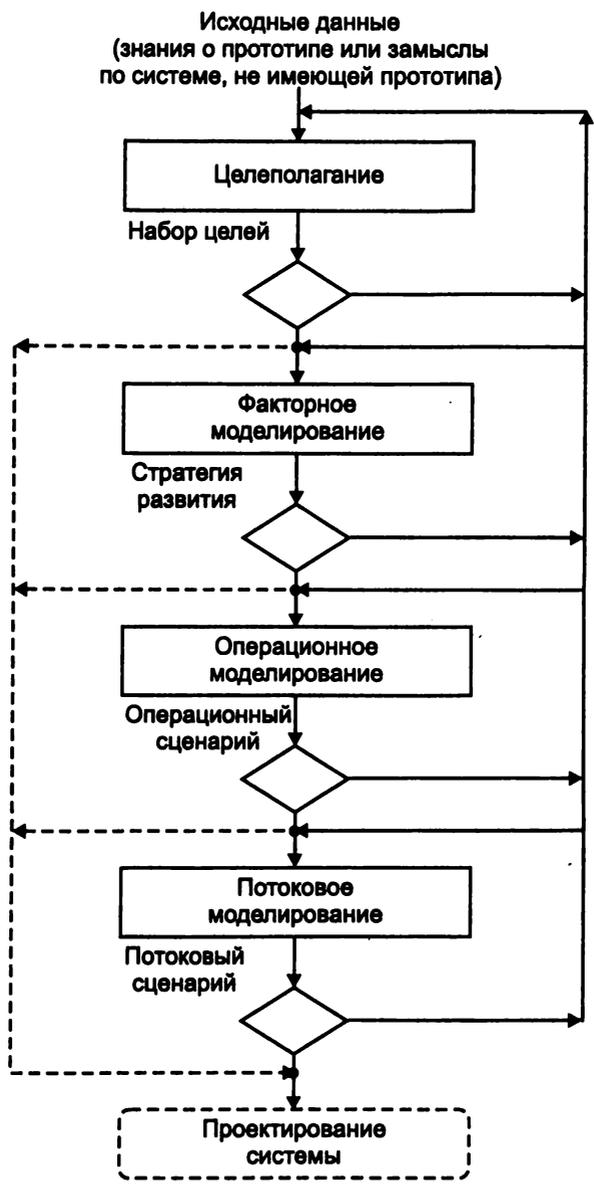


Схема предпроектного моделирования организационных систем

В большинстве случаев создаваемая (реформируемая) система строится не с «чистого листа», а на основе некоторого прототипа, в который вносятся необходимые изменения. Задача предпроектного моделирования заключается в том, чтобы разобраться в прототипе системы и принять решения по вносимым в нее изменениям. Эта задача решается путем построения цепочки модельных версий, где каждая последующая версия в определенном смысле совершенствует предыдущую. Возникает вопрос: что же взять в качестве первичной модельной версии?

Если прототип доступен и хорошо изучен, по нему собрана представительная статистика и определены пути возможного совершенствования, то в качестве первичной используется модель, отображающая усовершенствованный прототип. Если же разработчики системы не обладают достаточной информацией о свойствах прототипа и первичная модель должна помочь им определить эти свойства, а также пространство возможных изменений, то моделируется непосредственно прототип системы. В тех случаях, когда создается принципиально новая система, не имеющая прототипа, первичная модель отображает замыслы и намерения ее авторов.

Излагаемая авторами интегрированная методология предпроектного моделирования позволяет с помощью сравнительно простых и прозрачных моделей исследовать функционирование сложных систем. Для этого модель системы создается в виде многоуровневой иерархической структуры, где по отношению к каждому уровню применяются все вышеперечисленные этапы – *целеполагание, факторное, операционное, потоковое* моделирование.

Исходя из возможностей кратковременной памяти человека воспринимать и удерживать лишь ограниченный объем информации, вводятся ограничения на размерность моделей любого уровня. Для целеполагания это размеры дерева целей, для факторного моделирования – число факторов, для операционного моделирования – число операций, для потокового моделирования – число позиций, сопоставимое с числом операций (психологи рекомендуют, чтобы число компонентов моделей не превышало десяти).

Предпроектное моделирование проводят по уровням сверху вниз. Пусть с учетом введенных ограничений выстроен и исследован верхний уровень. Некоторые (или все) операции верхнего уровня могут соответствовать сложным процессам, требующим проведения своего целеполагания, факторного, операционного, потокового моделирования, либо части из них. Это служит осно-

ванием для введения модели следующего уровня. Каждый уровень прозрачен и моделируется независимо от других уровней. Таким образом реализуется принцип фрактальности (подобия).

Осмысливая предложенную парадигму интеграции этапов предпроектного моделирования организационных систем, приходим к следующему выводу: по мере движения по схеме (рисунок) сверху вниз снижается субъективность моделей, выражаемая экспертными оценками, и повышается их объективность (адекватность), основанная на применении точных математических методов. Ценой за адекватность (более детальное отображение функционирования системы) является усложнение моделей.

Отмеченная особенность предпроектного моделирования объясняется, на наш взгляд, тем обстоятельством, что первоначально эксперт (и аналитик) видит исследуемый предмет в свете личных представлений и предпочтений, отражающих его опыт. И лишь приобретая «критическую массу» знаний (в значительной мере в результате выполнения начальных этапов моделирования), он формирует более объективное структурированное представление о предмете и, возможно, изменит свое мнение.

### **Контрольные вопросы и упражнения**

1. Чем обусловлена потребность в предпроектном моделировании при создании и/или реформировании организационных систем?
2. Назовите последовательные этапы предпроектного моделирования и дайте краткую характеристику каждого этапа.
3. Почему сложные системы описывают на основе иерархических моделей? В чем заключается принцип подобия?
4. Как соотносится значимость субъективного и объективного факторов (экспертных оценок и строгих математических методов) на разных этапах предпроектного моделирования?

## ЦЕЛЕПОЛАГАНИЕ ПРИ ПРЕДПРОЕКТНОМ АНАЛИЗЕ ОРГАНИЗАЦИОННЫХ СИСТЕМ

### 1.1. Общая характеристика технологии целеполагания

Информационная технология выбора оптимального набора целей системы и уровней достижения этих целей при функционировании системы состоит из следующих этапов.

1. *Структурирование целей системы.* Выделяется главная (глобальная) цель, определяющая функционирование системы. Глобальная цель, которой присваивается нулевой уровень, разлагается (декомпозируется) на составляющие – цели первого уровня. Смысл разложения заключается в том, что выполнение цели определенного уровня обеспечивается выполнением *всех* целей нижестоящего уровня (для прозрачности описания число составляющих разлагаемой цели не должно превышать 4–5, что всегда возможно за счет увеличения числа уровней). Аналогично разлагаются цели первого уровня и т.д. до тех пор, пока не получатся неразлагаемые цели, которые будем называть локальными. Иерархическая структура целей системы представляется графом типа дерево, корень которого соответствует глобальной цели, а листья – локальным.

2. *«Взвешивание» целей системы.* Пусть число локальных целей, определенных на этапе 1, равно  $N$ , а число уровней достижимости каждой цели равно  $k$ . При этом пространство возможных решений, называемых многоцелевыми альтернативами (наборы значений уровней достижимости локальных целей), составляет  $N^k$ . Лобовой выбор в пространстве такой громадной размерности оптимальной альтернативы вряд ли возможен. Поэтому для эффективного выполнения целеполагания, во-первых, ранжируют локальные цели и отбирают из них наиболее значимые, умень-

шая тем самым величину  $N$ , и, во-вторых, снижают число уровней достижимости  $k$  до минимального разумного предела (например,  $k \leq 3$ ).

Для ранжирования целей каждая из них оценивается числовой величиной – весом цели. Процедуру определения весов, выполняемую при участии экспертов в определенной области, будем называть «взвешиванием» целей.

Универсальный способ «взвешивания» целей, называемый также методом анализа иерархий или методом Саати [12], последовательно применяется к двухъярусным фрагментам дерева целей, состоящим из корневой вершины и подчиненных ей вершин – листьев, в направлении сверху вниз. В начальном фрагменте, корневая вершина которого соответствует глобальной цели с заданным весом, эксперт попарно сравнивает все подчиненные цели (вершины нижнего яруса) на основе специальной шкалы, отражающей сравнительную значимость упорядоченной пары целей в качественном и количественном выражении. По результатам сравнения составляется система линейных уравнений, решение которой дает искомые веса подчиненных целей. Далее каждая подчиненная цель начального фрагмента дерева целей отождествляется с корневой вершиной нижестоящего фрагмента, и процедура повторяется вплоть до вычисления весов локальных целей системы.

Если структура целей определяется разложением целей вышестоящего уровня на множество нижестоящих целей, то может быть применен существенно более простой способ «взвешивания».

Для каждой  $i$ -й подчиненной цели фрагмента ( $i = 1, \dots, n$ ) эксперт непосредственно задает ее «долевой коэффициент»  $d_i$ , такой,

что  $0 < d_i < 1$ ,  $\sum_{i=1}^n d_i = 1$ , а вес цели  $w_i$  определяется произведением

$w_i = d_i \cdot w_k$ , где  $w_k$  – вес корневой вершины фрагмента.

Предлагая такой способ, который будем называть методом долевых коэффициентов, мы исходим из того, что для эксперта проще и естественнее сразу оценить долю составляющей (слагаемого) корневой цели фрагмента, чем оценивать сравнительную значимость составляющих, т.е. их пары (число составляющих на порядок меньше числа пар). Кроме того, отпадает необходимость в вычислительных манипуляциях, свойственных методу Саати.

В разд. 1.3 на примере иллюстрируются и сопоставляются оба метода «взвешивания» целей системы.

3. *Минимизация числа локальных целей системы.* Выбор наиболее значимых локальных целей системы осуществляется путем удаления наименее значимых из множества локальных целей, определенных на этапе 1 и «взвешенных» на этапе 2.

При минимизации числа локальных целей должно совместно выполняться несколько условий:

- необходимо учитывать взаимное влияние локальных целей, отображаемое матрицей – когнитивной картой [11, 17];
- суммарная степень достижимости удаляемых целей, имеющая численное выражение и определяемая на основе когнитивной карты, не должна превышать допустимого порогового значения;
- число удаляемых локальных целей должно быть максимально.

В разд. 1.4 дается формальное определение степени достижимости глобальной и локальных целей системы и предлагается эффективный алгоритм минимизации локальных целей.

4. *Классификация многоцелевых альтернатив.* Базовые локальные цели, оставшиеся после проведения минимизации на этапе 3, будем характеризовать уровнями достижимости, устанавливаемыми экспертами и выражаемыми словесными формулировками (типа «низкий», «средний», «высокий»). Словесным (качественным) значениям уровня сопоставляются количественные значения, например перечисленным выше – числа 1, 2, 3. Следует иметь в виду, что степень достижимости цели, используемая на этапе 3, и уровень достижимости – это разные понятия. Первое представляет собой функцию, определенную на когнитивной карте, второе – константу.

Множество  $A$  всех возможных наборов уровней достижимости базовых локальных целей образует пространство целеполагания, точки в котором называют многоцелевыми альтернативами (далее просто альтернативами).

Множество  $A$  на основе мнения экспертов разбиваем на непересекающиеся подмножества – классы альтернатив, определяемые ограничениями на потребляемые ресурсы, в первую очередь на финансовые и временные ресурсы. Среди классов выбираем приемлемые исходя из состояния ресурсов.

Классификация пространства целеполагания (разбиение множества  $A$  на классы) осуществляется методом Ларичева [7, 8], сочетающим компьютерные процедуры с обращениями к экспертам и позволяющим существенно уменьшить число таких обращений.

Метод базируется на бинарном отношении доминирования на множестве  $A$ : альтернатива  $a_i$  доминирует над альтернативой  $a_j$  ( $a_i, a_j \in A$ ), если все компоненты (уровни достижимости целей)  $a_i$  не меньше соответствующих компонент  $a_j$ , а значение по меньшей мере одной компоненты  $a_i$  строго больше, чем в  $a_j$ .

Суть метода Ларичева, который подробно обсуждается в разд. 1.5, заключается в следующем:

- компьютер просматривает множество  $A$ , для каждой альтернативы формирует подмножества доминирующих над ней и доминируемых ею альтернатив, на основе анализа этих подмножеств выбирает альтернативу  $a_i \in A$  для предъявления эксперту;

- эксперт классифицирует альтернативу  $a_i$  – относит ее к определенному классу. Если это «хороший» класс (приемлемый по ресурсам), то в него кроме  $a_i$  вносятся и все альтернативы, над которыми доминирует  $a_i$ , если класс «плохой» (неприемлемый), то вносятся все альтернативы, доминирующие над  $a_i$ ;

- классифицированные (отнесенные к классам) альтернативы удаляются из множества  $A$ , после чего процедура повторяется до тех пор, пока  $A$  не станет пустым.

5. *Определение в классе доминантных альтернатив.* Доминантными в классе являются такие альтернативы, над которыми не доминирует ни одна другая альтернатива (доминантные альтернативы называют также Парето-независимыми [8]).

Компьютерная программа определения доминантных альтернатив реализует последовательность шагов:

- формирует множество упорядоченных пар альтернатив;
- для каждой пары устанавливает, находится ли она в отношении доминирования;
- удаляет из класса все альтернативы, над которыми доминируют другие альтернативы.

6. *Определение в классе оптимальной альтернативы.* В каждой доминантной альтернативе класса уровни достижимости  $e_{ij}$ ,  $i = 1, \dots, n$ ;  $j = 1, \dots, k$ , локальных целей  $e_i$  интерпретируются как элементы нечеткого множества [3]  $E_i = (e_i, \mu_i)$ , где  $e_i = \{e_{ij}\}$ ,  $\mu_i : e_i \rightarrow [0, 1]$  – функция принадлежности, отображающая уровни  $e_{ij}$  в числа из интервала  $[0, 1]$ . Значение функции принадлежности  $\mu_i(e_{ij})$  трактуется как задаваемая экспертом вероятность достижения  $i$ -й локальной целью уровня  $e_{ij}$ .

Для каждой доминантной альтернативы  $a_q$ ,  $q = 1, \dots, b$ , вычисляется интегрированный показатель

$$G(a_q) = \sum_{i=1}^n \mu(e_{ij}) \cdot W(e_i), \quad (1.1)$$

где  $W(e_i)$  – вес  $i$ -й цели, определенный на этапе 2.

Оптимальной считается доминантная альтернатива с максимальным значением показателя  $G$ .

## 1.2. Структурирование целей системы

На рис. 1.1 показан пример дерева целей, определяющих реформирование производственной организационной системы. Цели обозначены буквой  $c$  с нижним индексом в виде набора цифр. Если цель  $c_i$  разлагается на составляющие, то последние обозначаются  $c_{i,1}, c_{i,2}, \dots$ . Глобальная цель (корень дерева) обозначена через  $c_0$ , в вершинах, инцидентных корню, нуль в индексе опускается. Укажем содержание обозначений целей:

Обозначение	Содержание
$c_0$	Увеличение прибыли
$c_1$	Снизить стоимость продукции
$c_2$	Улучшить потребительские свойства продукции
$c_{1,1}$	Снизить себестоимость выпускаемой продукции
$c_{1,2}$	Повысить рыночный спрос на продукцию
$c_{2,1}$	Улучшить дизайн изделий
$c_{2,2}$	Улучшить технические характеристики продукции (многофункциональность; надежность; срок службы; масса и габариты)
$c_{2,3}$	Улучшить гарантийное и последующее обслуживание продукции
$c_{1,1,1}$	Снизить транспортные расходы
$c_{1,1,2}$	Внедрить новые технологии в компании
$c_{1,1,3}$	Усовершенствовать управление и организацию труда
$c_{1,2,1}$	Увеличить расходы на рекламу
$c_{1,2,2}$	Добиться преимуществ перед конкурентами
$c_{2,3,1}$	Создание собственной сети сервисных центров
$c_{2,3,2}$	Осуществление сервисного обслуживания через другие фирмы

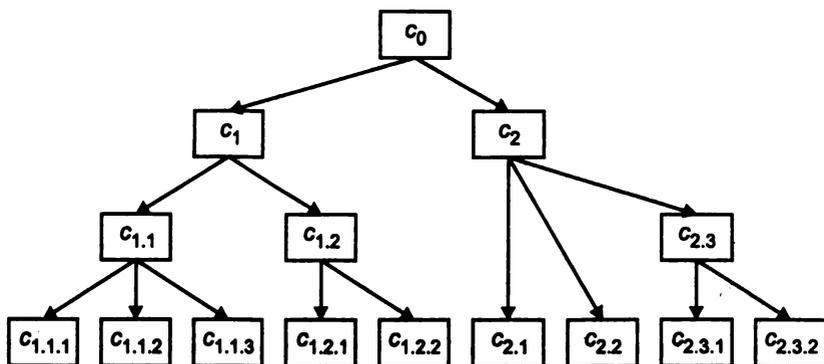


Рис. 1.1. Пример дерева целей

Выделим на дереве (при его обходе сверху вниз) двухуровневые фрагменты, состоящие из «родительской» вершины и совокупности инцидентных ей вершин – «потомков». Фрагменту, в котором родительской является «корневая» вершина дерева, присвоим нулевой ранг. Вершины нижнего уровня этого фрагмента являются родительскими для фрагментов первого ранга и т.д.

В примере на рис. 1.1 нулевой ранг имеет фрагмент  $\{c_0, c_1, c_2\}$ , первый ранг – фрагменты  $\{c_1, c_{1.1}, c_{1.2}\}$ ,  $\{c_2, c_{2.1}, c_{2.2}, c_{2.3}\}$ , второй ранг – фрагменты  $\{c_{1.1}, c_{1.1.1}, c_{1.1.2}, c_{1.1.3}\}$ ,  $\{c_{1.2}, c_{1.2.1}, c_{1.2.2}\}$ ,  $\{c_{2.3}, c_{2.3.1}, c_{2.3.2}\}$ .

### 1.3.

#### «Взвешивание» целей системы

Проиллюстрируем вначале метод «взвешивания» вершин дерева целей на основе *парных сравнений*. Сравнительная значимость целей оценивается числом баллов по шкале, согласованной с экспертами. Приведем пример такой шкалы:

Лингвистическое значение	Числовое (балльное) значение
Одинаковая значимость целей $c_i$ и $c_j$	1
Некоторое (слабое) преобладание значимости $c_i$ над значимостью $c_j$	3
Существенное (сильное) преобладание $c_i$ над $c_j$	5

Лингвистическое значение	Числовое (балльное) значение
Очевидное (очень сильное) преобладание $c_i$ над $c_j$	7
Абсолютное преобладание $c_i$ над $c_j$	9
Промежуточные значения между соседними значениями шкалы	2, 4, 6, 8

Для каждого фрагмента дерева целей, начиная с нулевого ранга, строится квадратная матрица  $R = \|r_{ij}\|$ .

Строки (столбцы) матрицы  $R$  соответствуют вершинам – «потомкам» фрагмента дерева. В верхней клетке крайнего левого столбца указывается заданный вес родительской вершины фрагмента (для корня дерева – глобальной цели  $c_0$  вес  $w_0 = 1$ ). На пересечении строки  $c_i$  и столбца  $c_j$  указывается значение  $r_{ij}$ , равное:

- 1, если  $c_i = c_j$ ;
- числу баллов  $b_{ij}$  по шкале сравнительной значимости (см. выше), если  $c_i$  более важна, чем  $c_j$ ;
- $1/b_{ij}$ , если  $c_i$  менее важна, чем  $c_j$ .

Пусть строки (столбцы) матрицы  $R$  соответствуют целям  $c_1, \dots, c_p$ , оцениваемым искомыми весами  $W_1, \dots, W_p$ , а родительская вершина имеет заданный вес  $W_q$ . При этом справедливо условие

$$W_q = \sum_{i=1}^p W_i. \quad (1.2)$$

Искомые веса  $W_i$ ,  $i = 1, \dots, p$ , являются решениями системы линейных уравнений:

$$\left\{ \begin{array}{l} W_1 = \frac{1}{p} \sum_{j=1}^p r_{1,j} W_j \\ \dots \\ W_{p-1} = \frac{1}{p} \sum_{j=1}^p r_{p-1,j} W_j \\ W_p = W_q - \sum_{i=1}^{p-1} W_i. \end{array} \right.$$

Для фрагмента нулевого ранга  $\{c_0, c_1, c_2\}$  в примере на рис. 1.1 матрица  $R$  представлена в табл. 1.1.

Матрица попарных сравнений для фрагмента нулевого ранга

$W_0 = 1$	$c_1$	$c_2$	Вес цели
$c_1$	1	3	0,75
$c_2$	1/3	1	0,25

Система линейных уравнений имеет вид

$$\begin{cases} W_1 = \frac{1}{2}(W_1 + 3W_2) \\ W_2 = 1 - W_1 \end{cases} \quad (1.3)$$

и дает решения:  $W_1 = 0,75$ ;  $W_2 = 0,25$ .

Аналогично определяются веса вершин для остальных фрагментов дерева целей.

При «взвешивании» вершин дерева целей методом *долевых коэффициентов* каждому фрагменту, начиная с нулевого ранга, сопоставляется двухстолбцовая таблица. Левый столбец помечен разлагаемой целью  $c_k$  (родительской вершиной фрагмента), строки помечены ее составляющими  $c_i$  (вершинами-потомками), в клетки вписываются долевые коэффициенты  $d_i$ . Веса  $W_i = d_i \cdot W_k$  указываются в правом столбце.

Набор двухстолбцовых таблиц для дерева целей (см. рис. 1.1) показан на рис. 1.2.

Сопоставляя два рассмотренных выше метода «взвешивания» целей применительно к разложению целого на части, отметим достоинства второго метода:

- при «взвешивании» фрагмента используется  $n - 1$  обращение

к эксперту вместо  $\frac{n \cdot (n - 1)}{2}$  при попарных сравнениях, где  $n$  – число вершин-потомков. При назначении долевых коэффициентов эксперт «держит» в своей кратковременной памяти не более 4–5 объектов (целей), что соответствует зоне психологического комфорта;

	$c_0(W_0 = 1)$		$c_1(W_1 = 0,75)$		$c_{1,1}(W_{1,1} = 0,56)$
$c_1$	0,75	$W_1 = 0,75 \cdot W_0 = 0,75$	0,75	$W_{1,1} = 0,56$	0,07
$c_2$	0,25	$W_2 = 0,25 \cdot W_0 = 0,25$	0,25	$W_{1,2} = 0,19$	0,73
			$c_{1,1}$		$c_{1,1,1}$
			$c_{1,2}$		$c_{1,1,2}$
					$c_{1,1,3}$
					$W_{1,1,1} = 0,04$
					$W_{1,1,2} = 0,41$
					$W_{1,1,3} = 0,11$
					$c_{2,3}(W_{2,3} = 0,08)$
$c_{1,2,1}$	0,16	$W_{1,2,1} = 0,03$	0,12	$W_{2,1} = 0,03$	0,90
$c_{1,2,2}$	0,84	$W_{1,2,2} = 0,16$	0,56	$W_{2,2} = 0,14$	0,10
			$c_{2,1}$		$c_{2,3,1}$
			$c_{2,2}$		$c_{2,3,2}$
			$c_{2,3}$		$W_{2,3,1} = 0,07$
					$W_{2,3,2} = 0,01$

Рис. 1.2. Набор двухстолбцовых таблиц для дерева целей, показанного на рис. 1.1

- упрощаются действия эксперта, в частности отпадает необходимость построения шкалы сравнительной значимости (см. выше) и работы с ней;

- упрощается вычислительный процесс (компьютерная программа) вследствие отсутствия необходимости в решении систем линейных уравнений.

## 1.4.

### Минимизация числа локальных целей системы

Пример когнитивной карты, отображающей взаимовлияние локальных целей, приведен в табл. 1.2. Строки и столбцы таблицы соответствуют локальным целям, слева от строки проставляется обозначение цели  $c_i$ , справа – ее вес  $W_i$ . На пересечении строки  $c_i$  и столбца  $c_j$  ( $i, j = 1, \dots, N$ ) записывается экспертная оценка  $+\alpha_{ij}$ , если достижение цели  $c_i$  усиливает достижение цели  $c_j$ , и оценка  $-\alpha_{ij}$ , если, наоборот, ослабляет. Коэффициент  $\alpha_{ij}$ , где  $0 \leq \alpha_{ij} \leq 1$ , отражает силу воздействия цели  $c_i$  на цель  $c_j$ , например, так, как показано на шкале оценки взаимовлияния целей.

Оценка  $\alpha_{ij}$  может не совпадать с точками на шкале, а находиться в интервале между ними (например, 0,6 – сила воздействия  $c_i$  на  $c_j$  занимает промежуточное значение между оценками «среднее» и «сильное»). Если цели не влияют друг на друга ( $\alpha_{ij} = 0$ ) либо характер их взаимодействия неизвестен, то клетку  $(c_i, c_j)$  таблицы оставляем пустой.

Лингвистическое значение	Числовая шкала оценки взаимовлияния целей
Отсутствие влияния	0
Очень слабое влияние	0,1
Слабое влияние	0,3
Среднее влияние	0,5
Сильное влияние	0,7
Абсолютное влияние	1,0

На когнитивной карте взаимовлияния целей введем количественные показатели – степень достижимости глобальной ( $c_0$ ) и локальной ( $c_j$ ) цели:

Когнитивная карта взаимовлияния локальных целей

	$c_{1.1.1}$	$c_{1.1.2}$	$c_{1.1.3}$	$c_{1.2.1}$	$c_{1.2.2}$	$c_{2.1}$	$c_{2.2}$	$c_{2.3.1}$	$c_{2.3.2}$	
$c_{1.1.1}$	+ 1,0		+ 0,3		+ 0,5					0,04
$c_{1.1.2}$	+ 0,2	+ 1,0	+ 0,7		+ 0,9	+ 0,5	+ 0,8	+ 0,1	+ 0,1	0,41
$c_{1.1.3}$	+ 0,7	+ 0,7	+ 1,0	+ 0,5	+ 0,9	+ 0,7	+ 0,9	+ 0,3	+ 0,3	0,11
$c_{1.2.1}$			+ 0,3	+ 1,0	+ 0,5	+ 0,5				0,03
$c_{1.2.2}$		+ 0,5	+ 0,5	- 0,2	+ 1,0		+ 0,3			0,16
$c_{2.1}$			+ 0,2	+ 0,5	+ 0,7	+ 1,0				0,03
$c_{2.2}$		+ 0,8	+ 0,7		+ 0,9		+ 1,0			0,14
$c_{2.3.1}$		+ 0,3	+ 0,5	+ 0,3	+ 0,7			+ 1,0		0,07
$c_{2.3.2}$			+ 0,5	- 0,3					+ 1,0	0,01

$$J(c_0) = \sum_{j=1}^N \sum_{i=1}^N \alpha_{ij} \cdot W_i, \quad (1.4)$$

$$J(c_j) = \frac{\sum_{i=1}^N \alpha_{ij} \cdot W_i}{J(c_0)} = \frac{\sum_{i=1}^N \alpha_{ij} \cdot W_i}{\sum_{j=1}^N \sum_{i=1}^N \alpha_{ij} \cdot W_i}. \quad (1.5)$$

Для определения степени достижимости глобальной цели  $J(c_0)$  следует в каждом  $j$ -м столбце табл. 1.2 просуммировать произведения экспертной оценки в клетке  $(i, j)$  на вес строки  $W_i$  и затем сложить полученные величины по всем столбцам.

Степень достижимости локальной цели  $J(c_j)$  определена как отношение указанной суммы произведений для  $j$ -го столбца к величине  $J(c_0)$ .

В рассматриваемом примере (см. табл. 1.2)  $J(c_0) = 3,55$ , а значения  $J(c_j)$  приведены в табл. 1.3.

Таблица 1.3

Выбранные локальные цели

$c_j$	$c_{1.1.1}$	$c_{1.1.2}$	$c_{1.1.3}$	$c_{1.2.1}$	$c_{1.2.2}$	$c_{2.1}$	$c_{2.2}$	$c_{2.3.1}$	$c_{2.3.2}$
$J(c_j)$	0,06	0,20	0,18	0,01	0,22	0,09	0,17	0,04	0,02
$e_j$	–	$e_1$	$e_2$	–	$e_3$	–	$e_4$	–	–

Множество всех локальных целей обозначим через  $C$ , подмножество удаляемых целей – через  $C^*$ , а его мощность – через  $|C^*|$ .

Степень достижимости подмножества целей  $C^* = \{c_{j_1}, \dots, c_{j_k}\}$  с учетом их взаимовлияния определяется соотношением

$$J(C^*) = J(c_{j_1}) + \dots + J(c_{j_k}). \quad (1.6)$$

Максимально допустимое значение  $J(C^*)$  обозначим через  $\Delta$ .

Задачу минимизации сформулируем следующим образом: найти  $C^* \subset C$  такое, чтобы совместно выполнялись условия:

$$J(C^*) \leq \Delta, \quad (1.7)$$

$$|C^*| = \max. \quad (1.8)$$

Минимизацию числа локальных целей будем проводить на основе следующего алгоритма.

1. В множестве  $C$  выбрать цель  $c_{j1}$  с минимальной степенью достижимости ( $J(c_{j1}) = \min$ ). Если таких целей более одной, то из них выбрать любую. Включить выбранную цель в подмножество  $C^*$  и увеличить его степень достижимости  $J(C^*) = J(C^*) + J(c_{j1})$ .

2. Проверить условие (1.7). Если оно выполняется, то удалить  $c_{j1}$  из  $C$  и перейти к шагу 1. Если условие (1.7) не выполняется ни для одной цели с  $J(c_{j1}) = \min$ , т.е. к  $C^*$  нельзя присоединить ни одной цели, то КОНЕЦ работы алгоритма (при выполнении условия (1.8)).

Работа алгоритма наглядно иллюстрируется в виде цепочки членов, соединенных стрелками. В числителе члена указывается состояние (состав) подмножества  $C^*$  на данном шаге минимизации, в знаменателе – соответствующая степень достижимости  $J(C^*)$ .

В примере (см. табл. 1.3) принято:  $\Delta = 0,25$  и цепочка имеет вид:

$$\begin{aligned} & \frac{\{c_{1.2.1}\}}{0,01} \rightarrow \frac{\{c_{1.2.1}, c_{2.3.2}\}}{0,03} \rightarrow \frac{\{c_{1.2.1}, c_{2.3.2}, c_{2.3.1}\}}{0,07} \rightarrow \\ & \rightarrow \frac{\{c_{1.2.1}, c_{2.3.2}, c_{2.3.1}, c_{1.1.1}\}}{0,13} \rightarrow \frac{\{c_{1.2.1}, c_{2.3.2}, c_{2.3.1}, c_{1.1.1}, c_{2.1}\}}{0,19} \end{aligned}$$

Цели, вошедшие в состав подмножества  $C^*$ , отмечаем в нижней строке табл. 1.3 прочерками, оставшиеся цели переобозначаем символами  $e_i$ ,  $i = 1, \dots, N^*$ , где

$$N^* = N - |C^*|. \quad (1.9)$$

В нашем примере результатом минимизации числа локальных целей системы является множество  $E = \{e_1, e_2, e_3, e_4\}$ ,

- где  $e_1$  обозначает внедрить новые технологии в компании;  
 $e_2$  – усовершенствовать управление и организацию труда;  
 $e_3$  – добиться преимуществ перед конкурентами;  
 $e_4$  – улучшить технические характеристики продукции.

## 1.5.

### Классификация многоцелевых альтернатив

Как уже говорилось, классификация заключается в распределении (сортировке) на классы решений множества альтернатив  $A = \{a_i\}$ ,  $i = 1, \dots, B$ , где  $a_i$  – значение набора целей  $(e_1 \dots e_{N^*})$ ;  $B = (N^*)^k$ ,  $k$  – число уровней цели.

Исходя из ограничений на финансовые ресурсы  $\Phi$  и временные ресурсы  $V$  множество  $A$  разобьем на классы решений  $K_1, \dots, K_R$ :

$$\bigcup_i K_i = A, K_{i_1} \cap K_{i_2} = \emptyset, \\ i, i_1, i_2 \in \{1, \dots, R\}, R \leq B, i_1 \neq i_2. \quad (1.10)$$

В примере  $k = 3$ , т.е. рассматриваются низкий, средний и высокий уровни достижения цели, кодируемые соответственно цифрами 1, 2, 3.

Имеем два класса решений:

$$K_1(\Phi \leq \Phi_0, V \leq V_0) \text{ и } K_2(\Phi > \Phi_0, V > V_0),$$

где  $\Phi_0, V_0$  – варьируемые пороговые значения соответственно капиталовложений и времени выполнения проекта.

«Худшая» альтернатива  $e_1 e_2 e_3 e_4 = 1111$  заведомо удовлетворяет введенным ограничениям на ресурсы, т.е. принадлежит классу  $K_1$ , а «лучшая» альтернатива 3333 согласно «ресурсным возможностям» относится к классу  $K_2$  (здесь и далее альтернативы обозначены 4-значными кодами). Другими словами, мы доопределяем классы таким образом, что «худшая» альтернатива (1111) принадлежит классу  $K_1$ , а «лучшая» (3333) принадлежит классу  $K_2$ .

Из классов  $K_1, K_2$  приемлемым является класс  $K_1$ .

Бинарное отношение доминирования, на основе которого классифицируется множество  $A$ , формально определяется следующим образом: альтернатива  $a_i = e_{i1}, \dots, e_{iN^*}$  доминирует над альтернативой  $a_j = e_{j1}, \dots, e_{jN^*}$ , если для любого  $g = 1, \dots, N^*$  имеет место  $e_{ig} \geq e_{jg}$  и существует по меньшей мере одно  $h = 1, \dots, N^*$  такое, что  $e_{ih} > e_{jh}$ . Например, альтернатива 1113 доминирует над альтернативой 1112, но 1121 не доминирует над 1113 и т.д.

Классификацию альтернатив осуществляем алгоритмом Ларичева [7, 8].

1. Для каждой альтернативы  $a_i \in A$  вычисляем коэффициенты  $D_i$  и  $D'_i$ , где  $D_i$  – число альтернатив, доминирующих над  $a_i$ ,  $D'_i$  – число альтернатив, над которыми доминирует  $a_i$ .

2. Из множества  $A$  выбираем те альтернативы, для которых  $|D_i - D'_i| = \min$ .

3. Из полученных в пункте 2 альтернатив выбираем те, для которых значение  $D_i + D'_i$  максимально.

4. Если пункт 3 оставил несколько альтернатив, то выбираем любую альтернативу  $a_p$ .

5. Классифицируем альтернативу  $a_p$  (предъявляем ее эксперту, который относит  $a_p$  к одному из классов) и после ее классификации экспертом исключаем  $a_p$  из множества  $A$ .

6. Если  $a_p$  отнесена к классу  $K_1$ , то из множества  $A$  исключаем также все альтернативы, которые доминируются альтернативой  $a_p$  (относим их к классу  $K_1$ ); если  $a_p$  отнесена к классу  $K_2$ , то из множества  $A$  исключаем все альтернативы, доминирующие по отношению к  $a_p$  (относим их к классу  $K_2$ ).

7. Если  $|A| > \emptyset$ , то возвращаемся к пункту 1, если  $|A| = 0$ , то КОНЕЦ работы алгоритма.

Процедура выбора альтернативы для предъявления эксперту с целью классификации проиллюстрирована на рис. 1.3, где овами обозначены множества альтернатив, квадратами или прямоугольниками – альтернативы. Как отмечалось выше, вначале для каждой альтернативы из  $A$  определяются множества доминирующих и доминируемых альтернатив (также определяются  $D_i$  и  $D'_i$ ). Далее из  $A$  выделяется множество альтернатив с минимальным значением  $|D_i - D'_i|$ , а затем из него выделяется множество с максимальным значением  $D_i + D'_i$ , из которого случайным образом выбирается  $a_p$  для классификации.

В нашем примере эксперту было задано 15 вопросов, т.е. классифицировано 79 альтернатив, из которых 72 были отнесены к классу  $K_1$ , удовлетворяющему ограничениям на финансовые и временные ресурсы.

Определение в классе доминантных альтернатив проводим на основе квадратной таблицы, строки и столбцы которой соответствуют альтернативам, принадлежащим выбранному классу.

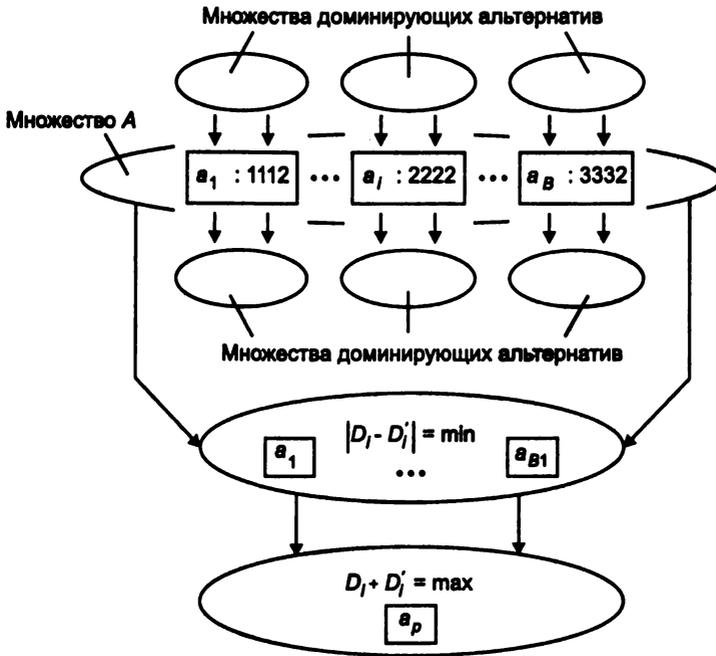


Рис.1.3. Процедура выбора альтернативы для предъявления эксперту

Примером может служить табл. 1.4. В строке  $a_i$  таблицы просматриваются все незаполненные клетки, и в клетку  $(a_p, a_j)$ ,  $i \neq j$ , вносятся знак «+», если альтернатива  $a_i$  доминирует над альтернативой  $a_j$ . При этом  $a_j$  не доминирует над  $a_i$ , и в клетку  $(a_p, a_i)$ , симметричную  $(a_p, a_j)$  относительно нисходящей диагонали, вносятся знак «-». Незаполненной считается клетка, в которой отсутствует как «+», так и «-». Применение знака «-» сокращает число рассматриваемых пар альтернатив при построчном заполнении таблицы (например, в направлении сверху вниз).

Доминантными являются те альтернативы  $a_p$ , в столбцах которых нет ни одного знака «+». Согласно табл. 1.4 это подмножество альтернатив  $A^* = \{3322, 3232, 3223, 2332, 2323, 2233\}$ .

Таблица 1.4

## Определение доминантных альтернатив

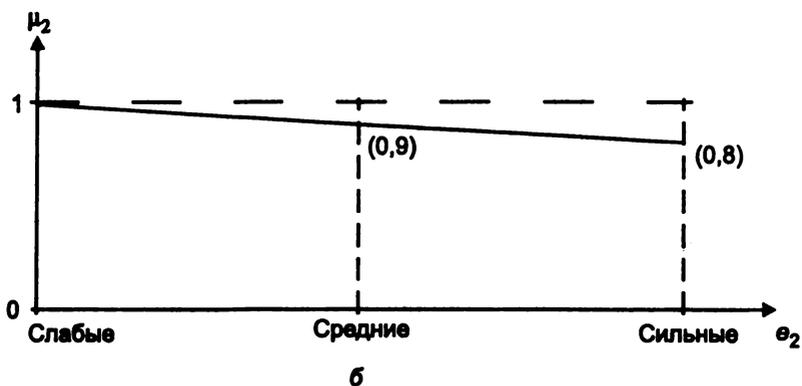
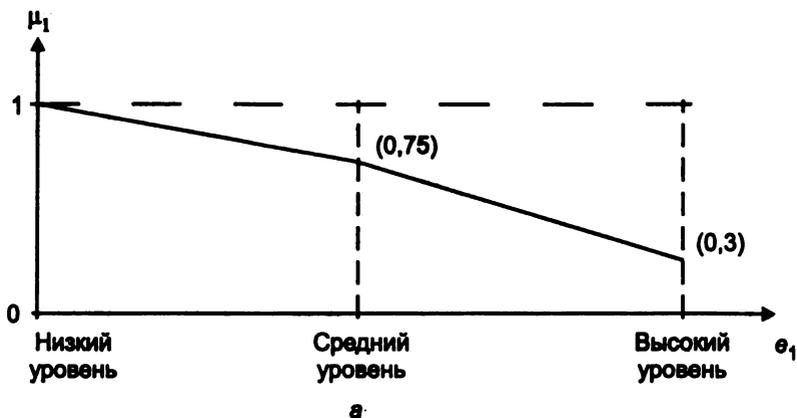
	3322	3232	3223	2332	2323	2233	3222	2232	2222	2231	1322	2223
3322	■						+					
3232		■						+				
3223			■						+			
2332				■						+		
2323					■						+	
2233						■			+			+
3222	-						■			+		
2232		-						■				
2222			-				-		■			
2231				-				-		■		
1322					-						■	
2223						-						■

## 1.6.

## Определение в классе оптимальной альтернативы

В выбранном классе  $K_1$  для каждой доминантной альтернативы  $a_q \in A^*$ , полученной на предыдущем этапе, вычисляем интегральный показатель  $G(a_q)$  согласно формуле (1.1). Для этого

каждой локальной цели  $e_i$ ,  $i = 1, \dots, n$  (см. разд. 1.4) сопоставляем экспертную функцию принадлежности  $\mu_i(e_{ij})$ , отображающую уровни  $e_{ij}$  достижимости цели, выраженные «качественными» (лингвистическими) значениями, в числа из интервала  $[0, 1]$ . Графики функций  $\mu_1, \dots, \mu_4$  для рассматриваемого примера даны на рис. 1.4.



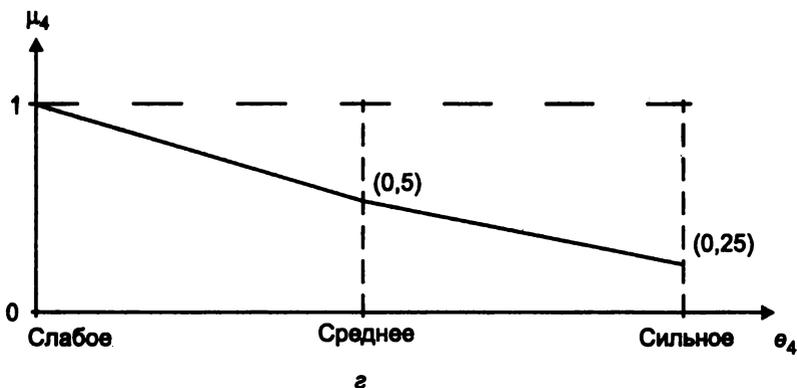
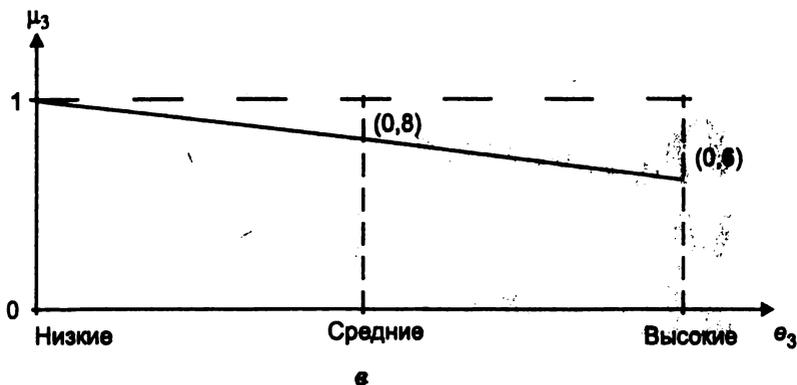


Рис. 1.4. Графики функций  $\mu_1, \dots, \mu_4$ :

- $a$  – внедрение новых технологий в компании;
- $b$  – усовершенствование управления и организации труда в компании;
- $e$  – преимущества перед конкурентами;
- $z$  – улучшение технических характеристик продукции

Далее для каждой доминантной альтернативы, входящей в множество  $A^*$ , рассчитывается показатель  $G$ , равный сумме произведений весов целей  $e_i, i = 1, \dots, 4$  (см. разд. 1.4) на соответствующие значения функций принадлежности  $\mu_i$  (см. рис. 1.4). Результаты даны в табл. 1.5.

Расчет показателя  $G$  для выбора оптимальной альтернативы

Альтернатива	Расчет $G$	Значение $G$
3322	$0,3 \cdot 0,41 + 0,6 \cdot 0,16 + 0,5 \cdot 0,14 + 0,9 \cdot 0,11$	0,39
3232	$0,3 \cdot 0,41 + 0,8 \cdot 0,16 + 0,25 \cdot 0,14 + 0,9 \cdot 0,11$	0,38
3223	$0,3 \cdot 0,41 + 0,8 \cdot 0,16 + 0,5 \cdot 0,14 + 0,8 \cdot 0,11$	0,41
2332	$0,75 \cdot 0,41 + 0,6 \cdot 0,16 + 0,25 \cdot 0,14 + 0,9 \cdot 0,11$	0,53
2323	$0,75 \cdot 0,41 + 0,6 \cdot 0,16 + 0,5 \cdot 0,14 + 0,8 \cdot 0,11$	0,56
2233	$0,75 \cdot 0,41 + 0,8 \cdot 0,16 + 0,25 \cdot 0,14 + 0,8 \cdot 0,11$	0,55

Выбираем альтернативу 2323, порождающую максимальное значение показателя  $G = 0,56$ . Эта альтернатива предусматривает:

- средний уровень внедрения новых технологий;
- высокий уровень управления и организации труда в компании;
- средний уровень преимуществ перед конкурентами;
- высокий уровень улучшения технических характеристик продукции;

Изложенная технология целеполагания может быть использована для решения широкого круга задач. Получаемый результат обладает большой степенью достоверности, так как вопросы, задаваемые эксперту, максимально упрощены и осуществляется постоянный контроль за непротиворечивостью ответов эксперта.

### Контрольные вопросы и упражнения

1. Самостоятельно сформулируйте глобальную цель, определяющую функционирование «учебной» организационной системы. Последовательно декомпозируйте глобальную цель на составляющие с получением иерархической структуры, изображаемой деревом. Выделите все локальные цели и все двухъярусные фрагменты дерева.
2. Поставьте себя в положение эксперта и методом долевых коэффициентов определите веса вершин построенного ранее дерева целей.

Проиллюстрируйте метод «взвешивания» на основе попарных сравнений (метод Саати) на примере фрагмента нулевого ранга дерева целей. Сравните оба метода «взвешивания».

3. В чем заключается идея, положенная в основу метода минимизации числа локальных целей системы? Составьте когнитивную карту взаимовлияния локальных целей для «взвешенного» дерева (вопрос 2). Определите по когнитивной карте степень достижимости глобальной цели и всех локальных целей. Сформулируйте задачу минимизации множества локальных целей и примените алгоритм минимизации для рассматриваемой вами учебной организационной системы.
4. Поясните, как формируется пространство целеполагания – множество альтернатив на базе набора отобранных локальных целей, как формируются классы альтернатив, определяемые ограничениями на потребляемые ресурсы, и как среди них выбирается предпочтительный класс? В чем заключается идея человеко-машинного метода Ларичева разбиения пространства целеполагания на классы?
5. Как определяются доминантные (Парето-независимые) альтернативы в выбранном классе пространства целеполагания? Как из доминантных альтернатив класса выбирается оптимальная альтернатива?

# ФАКТОРНОЕ МОДЕЛИРОВАНИЕ ОРГАНИЗАЦИОННЫХ СИСТЕМ

## 2.1.

### Принципы факторного моделирования

Факторное моделирование позволяет получить предварительный (грубый) прогноз развития системы на заданном временном интервале – динамику факторов, определяющих структуру системы и механизмы ее функционирования. Факторам сопоставляют переменные, значения которых вначале выражают качественно (в виде текстовых формулировок типа «слабый», «умеренный», «сильный» и т.д.) и затем переводят в числа – проектируют на количественную шкалу. Динамика фактора заключается в его возрастании или убывании.

Факторы подразделяются на целевые  $e_j, j = 1, \dots, n$ , и управляющие  $u_j, j = 1, \dots, m$ , влияющие на достижимость целевых. Методы выбора наиболее значимых целевых факторов и определения уровня их достижимости при ограничениях на ресурсы, основанные на экспертных оценках, обсуждались в гл. 1. Аналогичные методы применительно к управляющим факторам, включая определение их рейтинга, отбор по рейтингу, «взвешивание» выбранных факторов и т.д., исследуются в данной главе.

Временной интервал моделирования системы  $[\tau_0, \tau_k]$  представляет собой последовательность равномерно распределенных по оси времени моментов  $\tau$ , где  $\tau_0 \leq \tau \leq \tau_k$ . Состояние системы  $S$  в момент  $\tau$  определяется как набор числовых значений управляющих и целевых факторов:

$$S(\tau) = u_1(\tau), \dots, u_m(\tau), e_1(\tau), \dots, e_n(\tau). \quad (2.1)$$

При этом значение управляющего фактора  $u_j(\tau)$  задается алгебраической суммой

$$u_j(\tau) = \bar{u}_j(\tau) + \Delta u_j(\tau), \quad (2.2)$$

где  $\bar{u}_j(\tau)$  – значение фактора, устанавливаемое в результате внутренних взаимодействий в системе при переходе от предыдущего момента  $\tau - 1$  к  $\tau$ ;

$\Delta u_j(\tau)$  – приращение фактора, положительное или отрицательное, при внешнем управляющем воздействии на систему в момент  $\tau$ .

В частном случае  $\Delta u_j(\tau) = 0$ .

Для целевых факторов приращение, обусловленное непосредственно внешним воздействием, всегда нулевое.

Последовательность  $\Pi = S(\tau_0), \dots, S(\tau_k)$  состояний системы на интервале  $[\tau_0, \tau_k]$  будем называть прогнозом ее развития.

Динамика системы инициируется внешними управляющими воздействиями и взаимодействиями факторов между собой. Остановимся на этом подробнее.

Любой фактор, управляющий либо целевой, обозначим через  $x_i \in \{u_1, \dots, u_m, e_1, \dots, e_n\}$ . Пусть на фактор  $x_i$  непосредственно влияют факторы  $x_{j_1}, \dots, x_{j_g}$  так, как показано на рис. 2.1. Факторы соответствуют вершинам фрагмента ориентированного графа, взаимодействия факторов  $x_{j_h}x_i$ ,  $h = 1, \dots, g$ , – дугам графа, внешнее воздействие  $\Delta x_i$  – стрелке со свободным началом, ведущей в вершину  $x_i$ .

Каждая дуга  $x_{j_h}x_i$  помечается выражением

$$a_{j_h,i} = (z_{j_h,i}, \alpha_{j_h,i}), \quad (2.3)$$

где  $z_{j_h,i} \in \{+, -\}$  – знак взаимодействия факторов;

$\alpha_{j_h,i} \in [0, 1]$  – экспертная оценка силы взаимодействия.

Знак «+» говорит о том, что фактор  $x_{j_h}$  усиливает фактор  $x_i$  (при возрастании  $x_{j_h}$  возрастает  $x_i$ ), знак «-» говорит о том, что  $x_{j_h}$  противодействует  $x_i$  (при возрастании  $x_{j_h}$  убывает  $x_i$ ).

Сила взаимодействия  $a_{j_h,i}$  характеризуется качественно и количественно (например, так, как указано в разд. 1.4). Граф, фрагмент которого показан на рис. 2.1, называют взвешенным знаковым графом [11].

Динамика взвешенного знакового графа базируется на следующем механизме: если дуга  $x_{j_h}x_i$  помечена знаком «+», то при переходе от предыдущего момента  $\tau - 1$  к текущему моменту  $\tau$  фактор  $x_i$  получает по этой дуге положительное приращение  $a_{j_h,i} \times x_{j_h}(\tau - 1)$ ; если дуга помечена знаком «-», то отрицательное приращение; если дуга не помечена, то нулевое приращение. Прира-

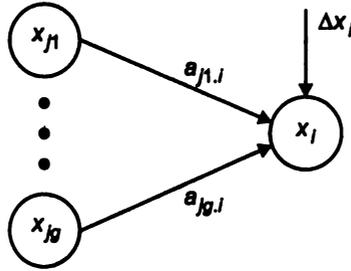


Рис. 2.1. Фрагмент взвешенного знакового графа

щения по всем входящим дугам, включая внешнее воздействие  $\Delta x_i$ , алгебраически суммируются. Формально этот механизм описывается уравнением

$$x_i(\tau) = \Delta x_i(\tau) + \sum_{h=1}^g a_{jh,i} \cdot x_{jh}(\tau-1), \quad (2.4)$$

где  $\Delta x_i = \begin{cases} \Delta u_i, & \text{если } x_i \in \{u_1, \dots, u_m\} \\ 0, & \text{если } x_i \in \{e_1, \dots, e_n\}. \end{cases}$

Пример взвешенного знакового графа приведен на рис. 2.2.

Граф представляют матрицей, называемой когнитивной картой [4, 5], строки и столбцы которой сопоставлены с факторами, а на пересечении строки  $x_i$  и столбца  $x_j$ ,  $i, j \in \{1, \dots, n + m\}$ , представлено значение  $a_{ij}$ . Графу, показанному на рис. 2.2, соответствует когнитивная карта, представленная в табл. 2.1.

Клетки  $x_i, x_j$  в табл. 2.1 оставляем незаполненными, если  $a_{ij} = 0$ .

Пример выполнения преобразования (2.4) при переходе  $\tau - 1/\tau$  на основе когнитивной карты (см. табл. 2.1) иллюстрируется схемой на рис. 2.3.

Согласно схеме вектор-столбец  $S(\tau-1)$  умножается на матрицу – когнитивную карту. В результате формируется новая матрица той же размерности. Значения, вписанные в каждый ее столбец, а также приращение соответствующего фактора, суммируются, что дает вектор-строку, транспонируемую в вектор-столбец  $S(\tau)$ . Таким образом формируется прогноз развития системы  $\Pi = S(\tau_0), \dots, S(\tau_k)$ .

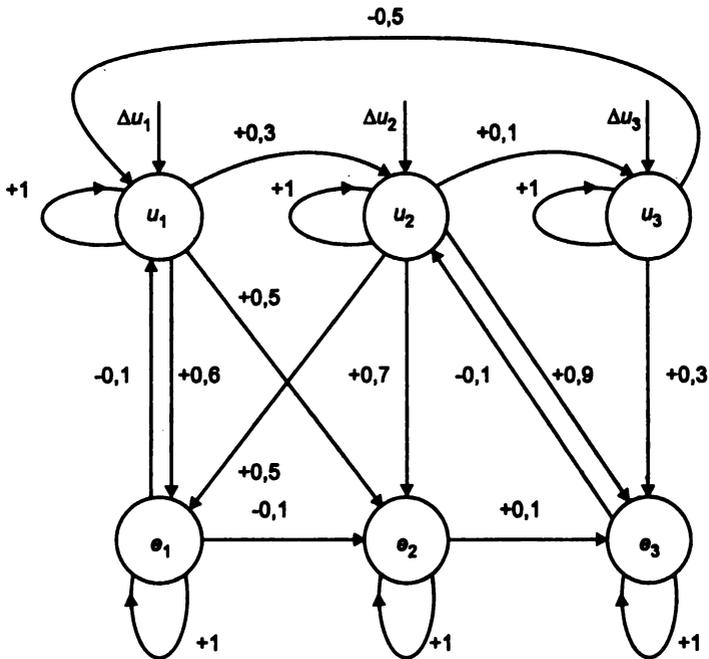


Рис. 2.2. Пример взвешенного знакового графа

Таблица 2.1

Когнитивная карта для графа, изображенного на рис. 2.2

	$u_1$	$u_2$	$u_3$	$e_1$	$e_2$	$e_3$
$u_1$	+1	+0,3		+0,6	+0,5	
$u_2$		+1	+0,1	+0,5	+0,7	+0,9
$u_3$	-0,5		+1			+0,3
$e_1$	-0,1			+1	-0,1	
$e_2$					+1	
$e_3$		-0,1			+0,1	+1

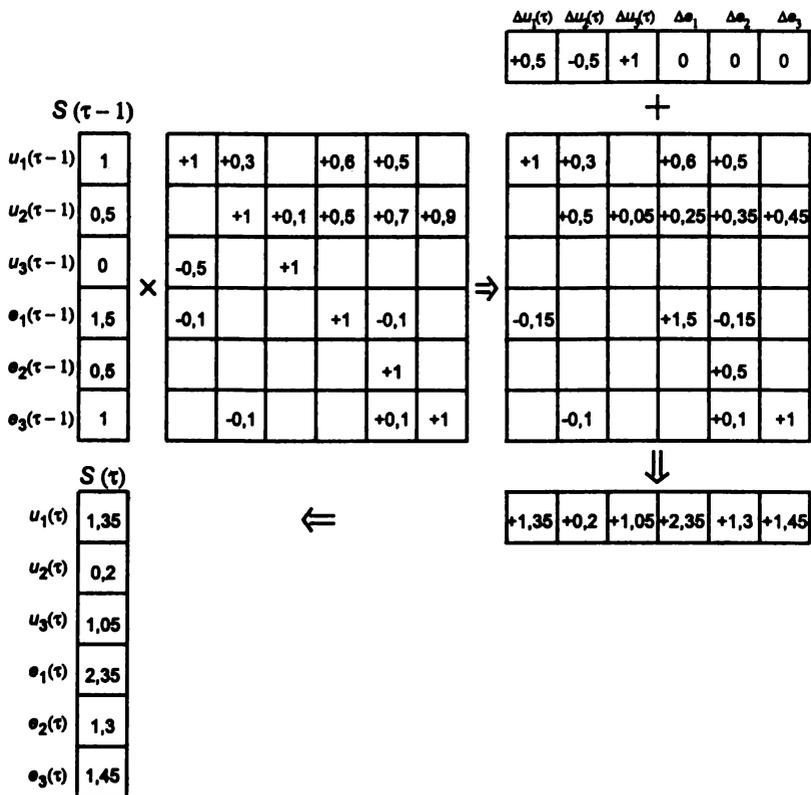


Рис. 2.3. Схема вычислений на основе когнитивной карты

Задачу факторного моделирования организационных систем сформулируем в следующем виде.

Пусть определено множество  $R = \{R_i\}$ ,  $i = 1, \dots, r$ , альтернативных стратегических решений (стратегий), которые могут быть реализованы в системе. Для каждого решения  $R_i$  составляем прогноз развития  $\Pi_i$ , характеризуемый начальным состоянием  $S_i(\tau_0)$ , когнитивной картой  $K_i$  и финальным состоянием  $S_i(\tau_k)$ . Требуется путем оценки прогнозируемых финальных состояний системы при различных стратегиях выбрать оптимальную стратегию.

Будем различать три варианта отношения между начальным состоянием (начальными условиями) и когнитивной картой при стратегиях  $R_i$ :

- все стратегии определяются единой когнитивной картой, но каждая своими начальными условиями;
- все стратегии определяются едиными начальными условиями, но каждая своей когнитивной картой;
- каждая стратегия определяется своими начальными условиями и своей когнитивной картой.

При выборе варианта учитывается, что вариативность начальных условий отображает реальные ситуации в системе, а вариативность когнитивных карт – разброс субъективного мнения экспертов.

Факторное моделирование организационных систем включает следующую последовательность шагов.

1. Выбор управляющих факторов, влияющих на достижимость целевых факторов, определенных на этапе целеполагания (гл. 1).

2. Формирование начальных условий (начальных значений целевых и управляющих факторов) и построение когнитивных карт.

3. Прогнозирование развития системы при различных стратегиях и выбор оптимальной стратегии.

Проведение факторного моделирования иллюстрируется в данной главе на примере производственной организационной системы, введенной в гл. 1. Прогнозируются три стратегии развития при единой когнитивной карте и различных начальных условиях.

## 2.2.

### Выбор управляющих факторов

Укажем управляющие факторы  $\Phi_1, \dots, \Phi_8$ , влияющие на достижимость целей  $e_1, \dots, e_4$  (см. разд. 1.4).

Обозначение	Содержание
$\Phi_1$	Ориентация на потребителя
$\Phi_2$	Лидерство руководителя
$\Phi_3$	Степень вовлечения работников
$\Phi_4$	Квалификация работников
$\Phi_5$	Системный подход к управлению организацией

Обозначение	Содержание
$\Phi_6$	Моральный климат в организации
$\Phi_7$	Принятие решений, основанное на фактах
$\Phi_8$	Взаимовыгодные отношения с поставщиками

Наиболее значимые из факторов  $\Phi_i$  устанавливаются по их рейтингу, который может быть определен двумя способами.

1. На основе экспертной оценки  $b_{ij}$  влияния управляющего фактора  $\Phi_i$  на цель  $e_j$  (с учетом уровня ее достижимости при ограничениях на ресурсы):

$$b_{i,j} = (z_{i,j} \cdot \beta_{i,j}), \quad (2.5)$$

где  $z_{i,j} \in \{+, -\}$ , знак «+» определяет приближение к цели, знак «-» удаление от нее;

$\beta_{i,j}$  – степень влияния фактора  $\Phi_i$  на цель  $e_j$ , выбираемая согласно эталонной шкале (см. разд. 1.4).

Интегральная оценка влияния фактора  $\Phi_i$  на все цели  $e_j$ ,  $j = 1, \dots, m$ , определяется по формуле

$$B_i = \sum_{j=1}^m (b_{i,j} \cdot W_j), \quad (2.6)$$

где  $W_j$  – вес цели  $e_j$  (в нашем примере веса указаны в табл. 1.2).

2. На основе экспертной оценки сравнительной значимости управляющих факторов (их попарного сравнения) и последующего вычисления весов факторов, определяющих их рейтинг, путем решения системы линейных уравнений (метод Саати [12]).

Второй способ связан с более сложными вычислениями, однако, возможно, упрощает работу экспертов, которые сравнивают «однородные понятия» (управляющие факторы).

В рассматриваемом примере применен первый способ определения рейтинга управляющих факторов. Результаты представлены в табл. 2.2. Факторы  $\Phi_i$  упорядочены по возрастанию их интегральной оценки, определенной согласно формуле (2.6). Рейтинги выражаются целыми положительными числами, наименьший рейтинг 1 имеет фактор  $\Phi_8$ , наибольший рейтинг 8 – фактор  $\Phi_1$ . Наиболее значимыми считаем те факторы, рейтинг которых не меньше 4. Эти факторы переобозначаем через  $u_1, \dots, u_5$  и используем далее при факторном моделировании.

Рейтинги управляющих факторов

Управляющие факторы	Вес цели				Интегральная оценка фактора	Рейтинг фактора	Выбранные управляющие факторы
	0,41	0,11	0,16	0,14			
	Выбранные цели и уровень их достижимости						
	$e_1 = 2$	$e_2 = 3$	$e_3 = 2$	$e_4 = 3$			
$\Phi_1$	+0,7	+0,8	+0,9	+0,9	0,64	8	$u_1$
$\Phi_2$	+0,5	+1,0	+0,3		0,36	4	$u_2$
$\Phi_3$	+0,3	+0,7	+0,7		0,31	3	–
$\Phi_4$	+0,7	+0,9	+0,8	+0,7	0,61	7	$u_3$
$\Phi_5$	+0,5	+0,8	+0,9	+0,3	0,48	5	$u_4$
$\Phi_6$	+0,7		+0,9	+0,7	0,53	6	$u_5$
$\Phi_7$	+0,3		+0,7	+0,5	0,30	2	–
$\Phi_8$			+0,3		0,05	1	–

Определим веса выбранных управляющих факторов  $u_1, \dots, u_5$ , используя для этого второй способ – метод попарного сравнения. Матрица попарных сравнений факторов  $u_i, i = 1, \dots, 5$ , приведена в табл. 2.3, а составленная на основе этой таблицы система линейных уравнений имеет вид (2.7) (исходим из того, что сумма весов  $\sum_{i=1}^5 W_i = 1$ ).

Таблица 2.3

Матрица попарных сравнений  
управляющих факторов

	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	Вес $W_i$
$u_1$	1	5	3	1	1	0,28
$u_2$	$1/5$	1	$1/5$	$1/7$	$1/7$	0,04
$u_3$	$1/3$	5	1	1	$1/3$	0,12
$u_4$	1	7	1	1	$1/3$	0,08
$u_5$	1	7	3	3	1	0,48

$$\begin{cases} W_1 = \frac{1}{5}(W_1 + 5W_2 + 3W_3 + W_4 + W_5); \\ W_2 = \frac{1}{5}(\frac{1}{5}W_1 + W_2 + \frac{1}{5}W_3 + \frac{1}{7}W_4 + \frac{1}{7}W_5); \\ W_3 = \frac{1}{5}(\frac{1}{3}W_1 + 5W_2 + W_3 + W_4 + \frac{1}{3}W_5); \\ W_4 = \frac{1}{5}(W_1 + 7W_2 + W_3 + W_4 + \frac{1}{3}W_5); \\ W_5 = 1 - W_1 - W_2 - W_3 - W_4. \end{cases} \quad (2.7)$$

Решение системы (2.7) дает веса  $W_i$ , указанные в правом столбце табл. 2.3.

Общая характеристика выбранных управляющих и целевых факторов производственной организационной системы с указанием количественных и качественных базовых начальных значений факторов (без учета приращений  $\Delta u_i(\tau_0)$ ,  $i = 1, \dots, 5$ ), а также весов факторов приведена в табл. 2.4.

Таблица 2.4

**Характеристика управляющих и целевых факторов организационной системы**

Фактор $u_i, e_j$	Содержание	Базовое начальное значение	Вес фактора $W_i$
$u_1$	Ориентация на потребителя	0,3 (слабая ориентация)	0,28
$u_2$	Лидерство руководителя	0 (отсутствие лидерства)	0,04
$u_3$	Квалификация работников	0,5 (средняя квалификация)	0,12
$u_4$	Системный подход к управлению организацией	0,1 (очень слабо внедряется)	0,08
$u_5$	Моральный климат в организации	0 (неудовлетворительный)	0,48
$e_1$	Внедрить в организации новые технологии	0 (не внедряются)	0,41
$e_2$	Усовершенствовать управление организацией	0,1 (незначительные усовершенствования)	0,11
$e_3$	Добиться преимуществ перед конкурентами	0 (отсутствие преимуществ)	0,16
$e_4$	Улучшить технические характеристики продукции	0,1 (незначительное улучшение)	0,14

### 2.3.

## Построение когнитивной карты и формирование начальных условий

Примем, что в рассматриваемой системе возможны три альтернативные стратегии развития:

- стратегия  $R_1$  – оставаться в секторе рынка  $A$  без существенного реформирования организации («оставить все, как есть»);
- стратегия  $R_2$  – переход в сектор  $B$  с коренным реформированием организации;
- стратегия  $R_3$  – оставаться в секторе  $A$  с одновременным наращиванием присутствия в секторе  $B$  с частичным реформированием организации.

Примем также, что все стратегии определяются единой когнитивной картой, едиными базовыми начальными значениями всех факторов (см. табл. 2.4), но каждая стратегия характеризуется своим приращением управляющих факторов в начальный момент  $\Delta u_i(\tau_0)$ ,  $i = 1, \dots, 5$ . Таким образом, каждой стратегии соответствуют свои начальные условия.

Когнитивная карта взаимовлияния факторов представлена в табл. 2.5, начальные приращения управляющих факторов  $\Delta u_i(\tau_0)$  при различных стратегиях – в табл. 2.6, начальные условия функционирования системы при различных стратегиях, полученные суммированием базовых начальных значений факторов и начальных приращений, даны в табл. 2.7.

Таблица 2.5

Когнитивная карта взаимовлияния факторов

$u_i, e_j$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$e_1$	$e_2$	$e_3$	$e_4$
$u_1$	+1,0		+0,5	+0,7	+0,7	+0,8	+0,9	+0,8	+0,9
$u_2$		+1,0		+0,3	-0,2	+0,7	+0,8	+0,7	+0,7
$u_3$	+0,3		+1,0	+0,7		+0,7	+0,7	+0,7	+0,7
$u_4$	+0,6		+0,5	+1,0	+0,7	+0,6	+0,9	+0,7	+0,7
$u_5$	+0,7		+0,5	+0,7	+1,0	+0,5	+0,7	+0,5	+0,5
$e_1$			+0,3	+0,3		+1,0	+0,5	+0,7	+0,7
$e_2$	+0,6		+0,7	+0,9	+0,5	+0,6	+1,0	+0,7	+0,7
$e_3$	+0,6	+0,5			+0,7			+1,0	
$e_4$	+0,5				+0,6			+0,8	+1,0

Таблица 2.6

Начальные приращения управляющих факторов при различных стратегиях

$\Delta u_i(\tau_0)$	Стратегия		
	$R_1$	$R_2$	$R_3$
$\Delta u_1(\tau_0)$	0	+0,6	+0,4
$\Delta u_2(\tau_0)$	0	+0,5	+0,5
$\Delta u_3(\tau_0)$	0	+0,2	+0,1
$\Delta u_4(\tau_0)$	0	+0,8	+0,6
$\Delta u_5(\tau_0)$	0	+0,7	+0,3

Таблица 2.7

Начальные условия функционирования системы при различных стратегиях

$u_i(\tau_0)$ $e_j(\tau_0)$	Стратегия		
	$R_1$	$R_2$	$R_3$
$u_1(\tau_0)$	0,3	0,9	0,7
$u_2(\tau_0)$	0	0,5	0,5
$u_3(\tau_0)$	0,5	0,7	0,6
$u_4(\tau_0)$	0,1	0,9	0,7
$u_5(\tau_0)$	0	0,7	0,3
$e_1(\tau_0)$	0	0	0
$e_2(\tau_0)$	0,1	0,1	0,1
$e_3(\tau_0)$	0	0	0
$e_4(\tau_0)$	0,1	0,1	0,1

## 2.4.

### Прогнозирование развития системы при различных стратегиях и выбор стратегии

Прогнозирование развития организационной системы на заданном временном интервале  $[\tau_0, \tau_k]$  по определенной стратегии  $R_v$ ,  $v = 1, \dots, r$ , проводится на основе (2.4) по схеме, иллюстрируемой на рис. 2.3. В результате прогнозирования определяется финальное состояние системы при стратегии  $R_v$ :

$$S(\tau_k, R_\nu) = u_1(\tau_k), \dots, u_m(\tau_k), e_1(\tau_k), \dots, e_n(\tau_k), \quad (2.8)$$

которое оценивается индексом результативности стратегии  $R_\nu$ :

$$I(R_\nu) = \sum_{j=1}^n (e_j(\tau_k) \cdot W_j), \quad (2.9)$$

вычисляемым на целевых факторах ( $W_j$  – вес фактора  $e_j$ ).

Оптимальной считается стратегия с максимальным значением индекса результативности.

Результаты прогнозирования развития производственной организационной системы при определенных выше стратегиях  $R_1$ ,  $R_2$ ,  $R_3$ , начальных условиях согласно табл. 2.7 и когнитивной карте согласно табл. 2.5 даны в табл. 2.8. Выбираем стратегию  $R_2$ , для которой  $I(R_2) = 2,44$ .

Таблица 2.8

Конечные значения факторов при различных стратегиях развития системы

Тенденция	Стратегия			Вес фактора
	$R_1$	$R_2$	$R_3$	
$u_1(\tau_k)$	0,6	2,2	1,1	0,28
$u_2(\tau_k)$	0	0,5	0,5	0,04
$u_3(\tau_k)$	0,8	2,0	1,5	0,12
$u_4(\tau_k)$	0,7	2,7	2,1	0,08
$u_5(\tau_k)$	0,3	1,9	1,3	0,48
$e_1(\tau_k)$	0,7	2,5	2,8	0,41
$e_2(\tau_k)$	0,8	3,1	2,4	0,11
$e_3(\tau_k)$	0,8	2,7	2,1	0,16
$e_4(\tau_k)$	0,9	3,8	2,2	0,17
$I(R_\nu), \nu = 1, 2, 3$	0,66	2,44	2,12	–

## Контрольные вопросы и упражнения

1. Приведите пример целевых и управляющих факторов организационной системы.
2. Постройте учебный «взвешенный» знаковый граф, описывающий взаимодействие факторов и влияние на них внешней среды. Представьте граф в виде матрицы – когнитивной карты.
3. Напишите уравнения, определяющие изменение значений факторов в последовательные моменты времени. Приведите матричную схему вычислений для перехода от момента  $\tau - 1$  к моменту  $\tau$ .
4. Сформулируйте задачу факторного моделирования организационной системы. Дайте пример реализуемых системой альтернативных стратегий.
5. Назовите три варианта отношений между начальным состоянием системы и описанием взаимодействия между факторами в виде когнитивных карт.
6. Укажите последовательность шагов при выборе оптимальной стратегии организационной системы.

# ОПЕРАЦИОННОЕ МОДЕЛИРОВАНИЕ ОРГАНИЗАЦИОННЫХ СИСТЕМ

### 3.1.

#### Операционный принцип моделирования функционирования систем

Глава посвящена разработке и исследованию (имитационному и аналитическому) сценариев функционирования организационных систем. Существуют различные толкования понятия «сценарий». Мы исходим из определения, согласно которому сценарий – это способ функционирования системы в виде множества действий (работ), выполняемых в определенном порядке и направленных на достижение определенных целей. Выбор целей и стратегии (пути) их достижения применительно к организационным системам были описаны соответственно в гл. 1 и 2, а гл. 3 является их естественным развитием с той особенностью, что если выбор целей и стратегии базируется в основном на экспертных оценках, т.е. имеет субъективный характер, то описание и анализ сценария функционирования основываются на строгом формальном (объективном) аппарате, который будет рассмотрен ниже.

В сценарии отображаются два уровня действий, выполняемых в системе: процессы и операции. Процессы направлены на достижение основных целей системы, они обладают высокой степенью автономности, функционируют преимущественно параллельно и могут вступать во взаимодействие. Процессы подразделяются на базовые и поддерживающие. Например, для производственной организационной системы к первым относятся материально-техническое обеспечение, изготовление продукции, продажа, доставка, обслуживание; ко вторым – НИОКР, имиджевое обеспечение, кадровое обеспечение, маркетинг, финансовая деятельность, руководство (управление).

Процессы состоят из операций, которые выполняются в определенном порядке, отражающем отношения во времени. Опе-

рации процесса могут выполняться как последовательно (конец одной операции совпадает с началом другой), так и параллельно; возможны альтернативный выбор последующей операции (в зависимости от складывающейся ситуации), взаимная синхронизация операций и другие отношения.

В зависимости от аспекта использования сценария при предпроектном анализе системы будем выделять целевые сценарии и сценарии логического управления. Целевой сценарий отображает планируемую динамику достижения целей системы в ходе выполнения операций, сценарий логического управления – планируемую динамику управляющих воздействий на систему, обеспечивающую требуемый порядок выполнения операций в зависимости от состояния системы и внешних ситуаций. В настоящей главе даны примеры сценариев обоих типов.

Поскольку сценарий функционирования системы определяется порядком выполнения операций, т.е. в этом контексте является операционным, то и моделирование на основе такого сценария называется операционным моделированием.

Операционное моделирование базируется на применении наглядных графоаналитических изобразительных средств в виде графов операций, объединяющих математический аппарат сетей Петри и алгебры событий, инициирующих переходы между операциями. В этой главе рассмотрены основы изобразительных средств (сети Петри и алгебра событий, включая равносильные преобразования описания событий), а также базовая операционная модель процесса и ее модификация, отображающая поведение процесса при возникновении нештатных ситуаций.

Предлагаются простая методика программирования на основе графов операций (с применением настраиваемых шаблонов) и методика проведения имитационных компьютерных экспериментов.

Граф операций [20] – это ориентированный граф с вершинами двух типов, сопоставленных с операциями и переходами, в котором дугами соединены только вершины разного типа. Вершины-операции дополнительно помечаются описанием динамики целевых и других переменных в целевых сценариях и описанием управляющих воздействий в сценариях логического управления, а вершины-переходы помечаются описанием событий. События, инициирующие переходы между операциями сценария, представляют собой композицию элементарных (базовых) событий, о которых подробно будет сказано позже. Такая графоаналитичес-

кая конструкция (без учета пометок вершин) соответствует модели «сеть Петри» [10]. Последняя является динамической моделью: в вершины операций (позиции сети Петри) помещаются маркеры, которые могут перемещаться между позициями в результате «срабатывания» переходов. В позиции либо нет маркера (операция не активизирована), либо есть маркер (операция активизирована, т.е. работает). Задается начальное размещение маркеров – начальная маркировка сети Петри. Срабатывание перехода в сети Петри определяется правилом: в каждой его входной позиции должен быть маркер, а в каждой выходной позиции маркер должен отсутствовать. В графе операций для срабатывания перехода требуется, кроме того, еще факт наступления события, помещающего этот переход.

Более подробное описание свойств сетей Петри и эффективных методов их анализа дается в разд. 3.2.

При операционном моделировании системы используются следующие типы переменных:

- целевые  $e_i \in E = \{e_1, \dots, e_n\}$ , отражающие степень достижения цели и принимающие лингвистические (а также числовые – «балльные») значения;

- операционные  $f_i \in F = \{f_1, \dots, f_m\}$ , где  $f_i$  – булевы переменные, принимающие единичное значение (ИСТИНА) при выполнении  $i$ -й операции и нулевое значение (ЛОЖЬ) во все остальные моменты;

- переходные  $t_i \in T = \{t_1, \dots, t_h\}$ , так же как и операционные, являющиеся булевыми переменными, принимающими единичное значение в момент срабатывания  $i$ -го перехода и нулевое значение во все остальные моменты;

- временные  $\tau_i \in \Theta = \{\tau_1, \dots, \tau_m\}$ , где  $\tau_i$  – продолжительность операции  $f_i$  (числовая переменная, принимающая целые неотрицательные значения);

- входные  $x_i \in X = \{x_1, \dots, x_k\}$ , внутренние  $y_i \in Y = \{y_1, \dots, y_p\}$ , выходные  $z_i \in Z = \{z_1, \dots, z_q\}$ , которые могут принимать как числовые, так и лингвистические значения.

События, инициирующие переходы на графе операций, могут зависеть от переменных всех типов.

Операционное моделирование проводится в форме диалога человека (системного аналитика) и компьютера в два этапа. На первом этапе человек составляет формальное описание сценария, а компьютер проверяет корректность этого описания и выдает

сообщения об ошибках. На втором этапе выполняется компьютерный имитационный эксперимент над разработанным сценарием, результатом эксперимента являются графики зависимости целевых, внутренних и выходных переменных (параметров системы) от единого системного времени.

При разработке *целевого сценария* будем исходить из того, что при выполнении операции те целевые переменные, на которые эта операция влияет, изменяются по линейному закону, скорость изменения будем задавать коэффициентом  $\alpha \in [-1, +1]$ . Конечное значение целевой переменной для операции-предшественника отождествляется с ее начальным значением для операции-последователя. Поэтому результат имитационного моделирования сценария для целевой переменной представляется непрерывной кусочно-линейной функцией.

При описании операций целевого сценария наряду с характеристикой изменения целевых переменных, как правило, задаются алгоритмы, формирующие значения внутренних и выходных переменных. В ходе имитационного эксперимента также строятся графики этих переменных на единой системной временной шкале.

Вопросы операционного моделирования на базе целевого сценария обсуждаются в разд. 3.3.

Построение *сценария логического управления* рассматривается применительно к распределенным организационным системам, представляющим собой совокупность автономно функционирующих взаимодействующих структурных компонентов. Каждый компонент реализует некоторый процесс, моделируемый соответствующим графом операций.

Взаимодействия между процессами (графами операций) в функциональном плане подразделяются на временные и обменные, а в социальном плане – на корпоративные и состязательные. Временные взаимодействия обеспечивают требуемую координацию работы процессов во времени. При обменных взаимодействиях [15, 16] процессы пересылают друг другу сообщения, которые в системе соответствуют финансовым, товарно-сырьевым, документационным и иным потокам.

Корпоративные взаимодействия процессов направлены на достижение общих целей системы (с которыми, естественно, должны быть согласованы собственные цели процессов), состязательные взаимодействия процессов исходят из собственных (эгоистических) целей.

При взаимодействиях процесс использует свои знания о состоянии других процессов. В случае корпоративных взаимодействий эти знания учитываются при формировании структуры событий, инициирующих переходы на графе операций. Модели составительных взаимодействий процессов базируются на теории игр [2].

Взаимодействие с внешней средой обеспечивается свойством реактивности – способностью процесса «поддерживать взаимодействие с окружающим миром, получая от него информацию и реагируя на нее своими действиями» [14]. Такими действиями могут быть выбор альтернативного маршрута на графе операций, блокирование функционирования графа и др. Реактивность обеспечивается структурой событий на графе операций.

Важное значение в организационной системе имеет ее архитектура. Будем различать децентрализованный и централизованный варианты архитектуры. В первом варианте функционируют только исполнительные процессы, реализуемые в структурных компонентах распределенной системы, причем функции принятия решений реализуются внутри исполнительных процессов. Во втором варианте в систему наряду с исполнительными процессами вводится процесс-координатор, реализуемый в центре управления, которому от исполнителей передаются (делегированы) некоторые (или все) функции принятия решений. Таким образом, децентрализованная архитектура организационной системы может быть преобразована в централизованную и, наоборот, централизованная архитектура преобразована в децентрализованную (путем передачи функций центра исполнителям). Такие преобразования полезны при поиске наилучшей архитектуры конкретных распределенных систем.

Операционное моделирование на базе сценария логического управления рассматривается в разд. 3.4.

## **3.2.**

### **Сети Петри – формальный аппарат операционного моделирования**

Операционную модель системы формируют в виде совокупности скоординированных между собой моделей процессов, представляемых сетями Петри (модификациями сетей Петри – графами операций). Изложим кратко принципы анализа сетей Петри, позволяющие установить корректность функционирования про-

цессов. Предмет рассмотрения – два взаимодополняющих метода, применение которых упрощает анализ в целом.

1. Процедура проверки условий корректности сети Петри – отсутствия в ней «тупиковых ситуаций» и «ловушек» (условие «живости») и запрет на помещение в позицию более одного маркера (условие «безопасности»). Классическая процедура анализа [10] базируется на построении так называемого дерева достижимости, которое может быть весьма громоздким и плохо обозримым. В разд. 3.2.1 описана процедура анализа сетей Петри на основе «ленты достижимости» [16] – линейной табличной формы представления множества состояний (маркировок), достижимых из начального состояния (маркировки) сети Петри. По сравнению с деревом достижимости лента более компактна (и поэтому лучше обозрима), формируется при помощи простой процедуры, не требующей к тому же компьютерного рисования.

2. Преобразование (редукция) сети Петри путем удаления из нее заведомо корректных (живых и безопасных) типовых фрагментов, осуществляемое итеративно, пока такие фрагменты выделимы. По завершении редукции либо отсутствует «остаток» (это говорит о том, что исходная сеть Петри корректна), либо формируется остаточная (редуцированная) сеть, по отношению к которой применяются упомянутую выше процедуру проверки условий корректности. Справедливо *утверждение*: исходная сеть Петри корректна тогда и только тогда, когда корректна редуцированная сеть. Чем меньше размерность редуцированной сети Петри по сравнению с размерностью исходной сети, тем эффективней применение метода редукции. Использование этого метода для целей анализа основано на иерархическом представлении сетей Петри, предложенном в [13] и развитом в [20].

Редукционный метод анализа сетей Петри излагается в разд. 3.3.2.

### 3.2.1.

#### Анализ сетей Петри на основе ленты достижимости

Напомним вначале, что *структура сети Петри* определяется как двудольный ориентированный граф (вершины графа разбиты на два подмножества  $P = \{p_i, i = 1, \dots, n\}$  и  $T = \{t_j, j = 1, \dots, k\}$ , смежными могут быть лишь вершины из разных подмножеств), в котором вершины  $p_i$  подмножества  $P$  называются позициями и

изображаются кружками, а вершины  $t_j$  подмножества  $T$  называются переходами и изображаются черточками. В позиции помещаются метки-маркеры, изображаемые жирными точками. Количество маркеров в позиции  $p_i$ , называемое ее маркировкой, выражается целым неотрицательным числом  $m_i$ . Наборы  $M_r = (m_1, \dots, m_n)$ ,  $r = 0, 1, \dots, N$ , определяют маркировки сети в целом. При задании сети Петри указывается ее начальная маркировка  $M_0$ . Пример структуры сети Петри с начальной маркировкой  $M_0 = (1, 0, 1, 0)$  приведен на рис. 3.1 [10].

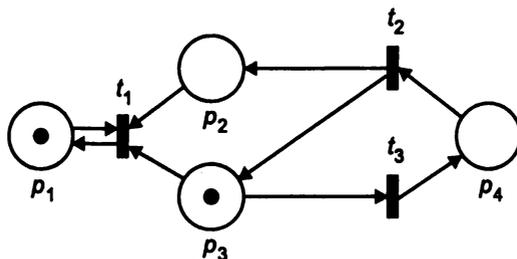


Рис. 3.1. Пример структуры сети Петри

*Поведение сети Петри* определяется изменением ее маркировки, обусловленным срабатыванием переходов. Без потери общности будем считать, что из позиции в переход (из перехода в позицию) ведет не более одной дуги. Переход подготовлен для срабатывания (активизирован), если в каждой его входной позиции есть по меньшей мере один маркер. При срабатывании перехода из каждой его входной позиции удаляется маркер, а в каждую выходную позицию маркер вносится. Срабатывание перехода происходит мгновенно. Если в сети в какой-нибудь маркировке активизированы два или более перехода, то может сработать только один из них. Выбор срабатывающего перехода осуществляется случайно, в этом смысле сеть Петри – недетерминированная модель. Для детерминизации сети Петри необходимо дополнительно ввести механизм выбора – управление сетью. Один из вариантов такого управления реализован в графе операций.

Сеть Петри является также асинхронной моделью, она работает не в физическом, а в логическом (дискретном) времени, определяемом частичной упорядоченностью событий (переходов).

В примере на рис. 3.1 в начальной маркировке  $M_0 = (1,0,1,0)$  активизирован только переход  $t_3$ . В результате его срабатывания маркер удаляется из позиции  $p_3$  и вносится в позицию  $p_4$  – маркировка  $M_0$  изменяется на  $M_1 = (1,0,0,1)$ , в которой активизирован только переход  $t_2$ . Срабатывание  $t_2$  приводит к маркировке  $M_2 = (1,1,1,0)$ , в которой активизированы переходы  $t_1$  и  $t_3$ . Если из начальной маркировки  $M_0$  неограниченное число раз повторять циклическую последовательность срабатываний  $t_3t_2$ , то и число маркеров в позиции  $p_2$  будет неограниченно возрастать. Этот факт отображается значением маркировки позиции  $p_2$ :  $m_2 = \omega$ , где  $\omega$  – символ бесконечности. Для  $\omega$  и любого заданного числа  $a$  определены условия [10]:

$$\begin{aligned} \omega + a &= \omega, a < \omega, \\ \omega - a &= \omega, \omega \leq \omega. \end{aligned} \quad (3.1)$$

Поведение сети Петри моделируется лентой достижимости (пример ленты достижимости для сети Петри на рис. 3.1 приведен в табл. 3.1).

В строки ленты вносятся соответственно обозначение маркировки  $M_r = (m_1, \dots, m_n)$ ,  $r = 0, 1, \dots, N$ , сети Петри, значения маркировок  $m_i$  ( $i = 1, \dots, n$ ) ее позиций  $p_i$ , множество  $\{t_r\}^{\text{акт}}$  переходов, активизированных в маркировке  $M_r$ . В нижней строке указывается переход, выбранный для срабатывания в данной маркировке, либо пометка «тупик/цикл». Пометка «тупик» соответствует случаю, когда в  $M_r$  не активизирован ни один переход, пометка «цикл» означает, что в ленте уже отражено срабатывание сети Петри из  $M_r$  по всем активизированным переходам.

Таблица 3.1

Пример ленты достижимости

$M_r$	$M_0$	$M_1$	$M_2$	$M_3$	$M_2$	$M_4$	$M_2$
$m_1$	1	1	1	1	1	1	1
$m_2$	0	0	$\omega$	$\omega$	$\omega$	$\omega$	$\omega$
$m_3$	1	0	1	0	1	0	1
$m_4$	0	1	0	0	0	1	0
$\{t_r\}^{\text{акт}}$	$\{t_3\}$	$\{t_2\}$	$\{t_1, t_3\}$	–	$\{t_1, t_3\}$	$\{t_2\}$	$\{t_1, t_3\}$
$t$	$t_3$	$t_2$	$t_1$	Тупик	$t_3$	$t_2$	Цикл

Столбцы ленты отождествляются с маркировками, лента последовательно заполняется слева направо, начиная с начальной маркировки  $M_0$ .

Идея алгоритма построения ленты достижимости заключается в следующем. Крайний справа столбец (маркировка), порождаемый в ходе построения ленты достижимости, называется граничным, а порожденные на предыдущих шагах столбцы (расположенные слева) – внутренними.

Для граничного столбца  $M_r$  определяется наличие в нем строк  $m_i = \omega$ . Для этого по ленте проходим путь (последовательность столбцов), ведущий из  $M_0$  в  $M_r$ . При формировании пути двигаемся справа налево – из конца в начало. Если на пути появляется внутренний столбец с маркировкой  $M_g$ , встречающийся в ленте  $h \geq 2$  раз ( $h$  – число переходов, активизированных в  $M_g$ ), то из  $h$ -го вхождения внутреннего столбца  $M_g$  «перескакиваем» к его 1-му вхождению. Если в найденном пути  $M_0 \rightarrow M_r$  присутствует такой внутренний столбец  $M_d$ , что для упорядоченной пары  $\langle M_r = (m_{r1}, \dots, m_{rn}), M_d = (m_{d1}, \dots, m_{dn}) \rangle$  по всем позициям  $p_i$  будет иметь место  $m_{ri} \geq m_{di}$  ( $M_r$  «покрывает»  $M_d$ ), то в граничном столбце  $M_r$  в тех строках, где имеет место строгое неравенство  $m_{ri} > m_{di}$ , значение  $m_{ri}$  заменяем на  $\omega$ . Активизированные переходы в столбце  $M_r$  определяются с учетом условия (3.1).

Если граничная маркировка  $M_r$  не содержит пометок «тупик» или «цикл», то определяем следующую маркировку  $M_s$ , в которую попадает сеть Петри при срабатывании выбранного перехода  $t_j$  (вписанного в нижнюю клетку столбца  $M_r$ ). Вводим в ленту достижимости граничный столбец  $M_s$ , а столбец  $M_r$  объявляем внутренним. Заполнение столбца  $M_s$  осуществляется следующим образом: содержимое тех клеток предыдущего внутреннего столбца  $M_r$ , которые соответствуют входным (выходным) позициям перехода  $t_j$ , уменьшается (увеличивается) на единицу, содержимое остальных клеток переносится без изменения.

Построение ленты достижимости завершаем, когда в нее нельзя ввести граничный столбец без пометок «тупик» или «цикл».

Описанный алгоритм проиллюстрируем на примере табл. 3.1. В маркировке  $M_2$  активизированы два перехода –  $t_1$  и  $t_3$ , в маркировке  $M_3$  – ни одного, в остальных маркировках – ровно по одному переходу (т.е. выбора как такового нет). Столбец  $M_3$  помечаем как тупиковый, а граничному столбцу  $M_2$  приписываем пометку «цикл». Последнее обусловлено тем, что граничной мар-

кировке  $M_2$  дважды предшествует внутренняя маркировка  $M_2$ , при первом ее появлении срабатывает переход  $t_1$ , при втором –  $t_3$ . Граничная маркировка  $M_2$  дублирует одну из соответствующих внутренних маркировок и порождает аналогичное ей последующее поведение сети.

Покажем, как вводится символ  $\omega$  и определяется заполнение столбцов ленты. Пусть в ходе построения ленты достижимости граничным был столбец, соответствующий первому появлению маркировки  $M_2$  (лента состояла из трех столбцов). В маркировку  $M_2 = (1, 1, 1, 0)$  мы перешли из маркировки  $M_1 = (1, 0, 0, 1)$  в результате срабатывания перехода  $t_2$  с входной позицией  $p_4$  и выходными  $p_2, p_3$  (см. рис. 3.1). Поэтому в столбце  $M_2$ :  $m_4 = 1 - 1 = 0$ ,  $m_2 = m_3 = 0 + 1 = 1$ , значение  $m_1 = 1$  позиции  $p_1$  перенесено без изменений. Но граничная маркировка  $M_2$  совпадает с начальной  $M_0 = (1, 0, 1, 0)$  по позициям  $p_1, p_3, p_4$ , а по позиции  $p_2$  в  $M_0$   $m_2 = 0$ , а в  $M_2$   $m_2 = 1$ . Следовательно, граничная маркировка  $M_2$  покрывает начальную маркировку  $M_0$ , поэтому в столбце  $M_2$  значение  $m_2 = 1$  заменяем на  $m_2 = \omega$  (перечеркиваем 1 и справа от нее вписываем  $\omega$ ).

Из  $M_2 = (1, \omega, 1, 0)$  при срабатывании  $t_1$  (входные позиции перехода  $t_1$ :  $p_1, p_2, p_3$ , выходная позиция  $p_1$  – см. рис. 3.1) попадаем в маркировку  $M_3 = (1, \omega, 0, 0)$ :  $m_1 = 1 - 1 + 1 = 1$ ,  $m_2 = \omega - 1 = \omega$ ,  $m_3 = 1 - 1 = 0$ ,  $m_4 = 0$  (значение  $m_4$  не изменяется). Далее лента заполняется подобным образом.

*Анализ свойств сети Петри* будем проводить по ее ленте достижимости. Напомним некоторые определения, используемые при анализе [10]:

- сеть Петри называется *K-ограниченной*, если существует постоянное число  $K$ , такое, что число маркеров в любой позиции сети не превышает  $K$ . Если  $K = 1$ , то сеть Петри называется *безопасной*;
- маркировка  $M_j$  достижима из маркировки  $M_i$ , если существует последовательность переходов, переводящая сеть из  $M_i$  в  $M_j$ ,  $i, j = 0, \dots, N$  (за счет использования символа  $\omega$  число маркеров сети Петри всегда конечно);
- маркировку  $M_p$  достижимую из начальной маркировки  $M_0$ , называют *целевой*, если попадание в  $M_p$  интерпретируется как завершение функционирования сети. На множестве маркировок  $M = \{M_0, M_1, \dots, M_N\}$  задается подмножество  $M^c$  целевых маркировок ( $M^c$  может совпадать с  $M$  либо быть пустым);

- сеть Петри называется «живой», если в ней из любой маркировки, достижимой из  $M_0$ , достижима по меньшей мере одна целевая маркировка.

Справедливо очевидное *утверждение*: сеть Петри является «живой», если в ней отсутствуют достижимые из  $M_0$  тупиковые нецелевые маркировки и отсутствуют циклы-«ловушки», образованные только из нецелевых маркировок.

Сеть Петри как инструмент операционного моделирования является корректной, если она «живая» и безопасная. И то и другое сравнительно просто распознается на ленте достижимости:

свойство безопасности проявляется в том, что все столбцы ленты заполнены только единицами и нулями (отсутствие символов  $\omega$  говорит о том, что сеть ограниченная);

свойство «живости» проявляется отсутствием в ленте столбцов с пометкой «тупик», соответствующих нецелевым маркировкам, и отсутствием таких циклов, в которых все маркировки нецелевые и во вторую снизу клетку каждого столбца вписан один переход, т.е. в маркировке активизирован только один переход (цикл – «ловушка»).

Сеть Петри на рис. 3.1 согласно ее ленте достижимости (см. табл. 3.1) небезопасная (и даже неограниченная из-за наличия символа  $\omega$ ) и «неживая». Пусть целевыми в сети являются маркировки  $M_2, M_4$ . Но тогда тупиковая маркировка  $M_3$ , достижимая из  $M_0$ , нецелевая (при попадании в  $M_3$  сеть «зависнет»). Заметим, что цикл  $(M_2, M_4)$ , инициируемый последовательностью переходов  $(t_3, t_2)$ , не является «ловушкой», так как в маркировке  $M_2$  активизированы два перехода.

### 3.2.2.

#### Редукция сети Петри

Метод редукции сети Петри основывается на следующих понятиях [13,20]:

- переход сети Петри называется автоматным, если он имеет не более одной входной и не более одной выходной дуги. В противном случае он называется неавтоматным;

- блоком на базе позиций  $p_{i1}, \dots, p_{ik}$  ( $i_1, \dots, i_k \in \{1, \dots, n\}$ ) называется фрагмент сети Петри, который содержит все указанные позиции (базовые позиции) и все смежные (связанные) с ними переходы. Блок обозначается через  $B_j(p_{i1}, \dots, p_{ik})$ ,  $j = 1, \dots, e$  – номер

блока. Переходы блока подразделяются на внутренние, которые связаны только с базовыми позициями, и на внешние, связанные не только с базовыми позициями блока, но и с другими позициями сети. Входные внешние переходы называют источниками блока, выходные – стоками;

- блок называется автоматным (А-блоком), если:  
он содержит более одной базовой позиции;

все его внутренние переходы – автоматные, а все внешние – неавтоматные, причем каждый внешний переход связан ровно с одной базовой позицией;

в начальной маркировке сети либо все базовые позиции блока пусты, либо в одной из базовых позиций содержится один маркер;

- блок называется блоком с параллелизмом (П-блоком), если:  
он содержит более одной базовой позиции;

в нем отсутствуют внутренние переходы, а каждый внешний переход связан со всеми базовыми позициями;

в начальной маркировке сети либо все базовые позиции блока пусты, либо в каждой из них содержится по одному маркеру.

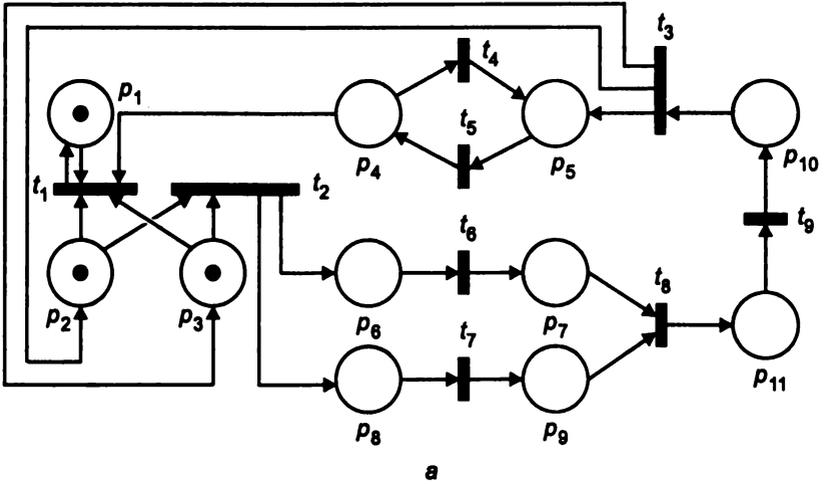
Изложенные понятия иллюстрируются примером на рис. 3.2. В сети Петри, изображенной на рис. 3.2, а, переходы  $t_1, t_2, t_3, t_8$  – неавтоматные, а остальные переходы – автоматные. Сеть содержит блоки  $B_1, \dots, B_5$ , показанные на рис. 3.2, б. Из них  $B_1(p_2, p_3)$  является П-блоком, а  $B_2(p_4, p_5), B_3(p_6, p_7), B_4(p_8, p_9), B_5(p_{10}, p_{11})$  являются А-блоками.

*Алгоритм редукции сети Петри* предусматривает следующие шаги.

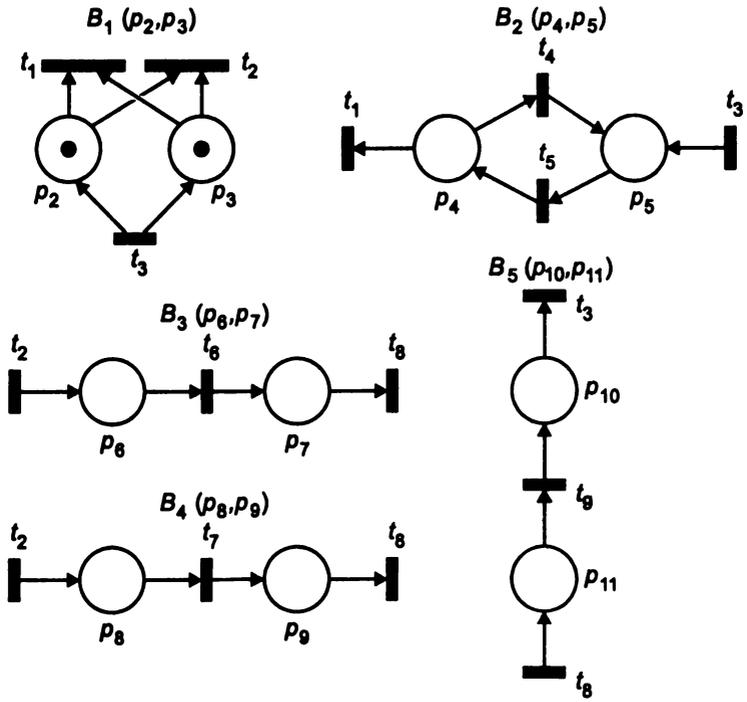
1. В исходной сети выделяются все А-блоки и П-блоки.

2. Каждый блок  $B_j(p_{i1}, \dots, p_{ik})$  замещается позицией  $B_j$ . Для этого из исходной сети удаляются базовые позиции блока  $p_{i1}, \dots, p_{ik}$  и все его внутренние переходы. В сеть вводится «новая» позиция  $B_j$ , которая связывается дугой с каждым внешним переходом блока (входящей дугой с источником и выходящей дугой со стоком). В начальной маркировке сети введенную позицию  $B_j$  оставляем пустой, если все базовые позиции блока были пусты, в противном случае в позицию  $B_j$  помещаем маркер.

К полученной в результате сети применяем шаги 1, 2 и продолжаем процесс редукции до тех пор, пока сеть не будет содержать ни одного А-блока и П-блока.



a



b

Рис. 3.2. Исходная сеть Петри (а) и содержащиеся в ней блоки (б)

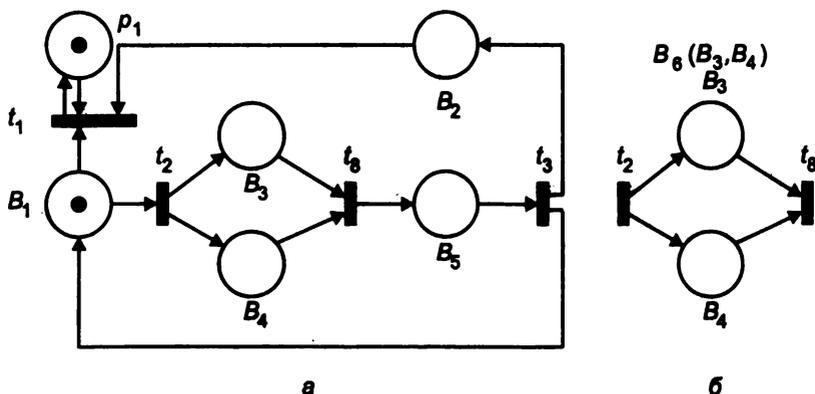
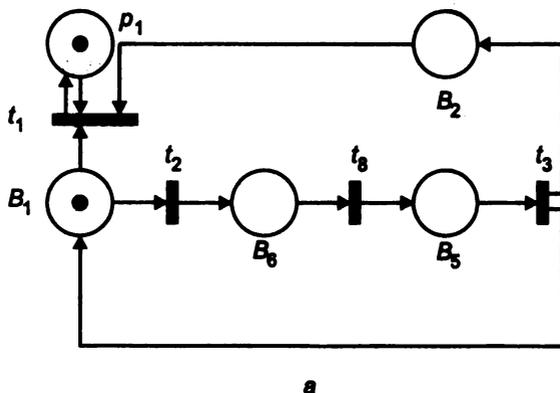


Рис. 3.3. Результат первой итерации алгоритма редукции (а) и содержащийся в ней блок (б)

В рассматриваемом примере (см. рис. 3.2) замещение позициями блоков  $B_1, \dots, B_5$  порождает сеть (рис. 3.3, а), которая содержит П-блок  $B_6(B_3, B_4)$ , показанный на рис. 3.3, б.

К сети на рис. 3.3, а повторно применяем шаги 1, 2 и получаем сеть, показанную на рис. 3.4, а, содержащую А-блок  $B_7(B_5, B_6)$  (рис. 3.4, б). Замещение позицией  $B_7$  этого блока дает редуцированную сеть (рис. 3.5), которая с точностью до обозначения и размещения позиций и переходов совпадает с сетью Петри (см. рис. 3.1), исследованной ранее (см. разд. 3.2.1).



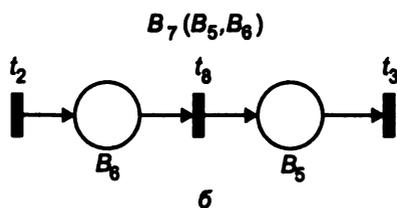


Рис. 3.4. Результат второй итерации (а) и содержащийся в ней блок (б)

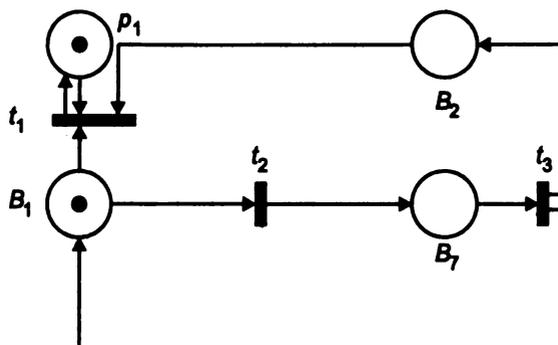


Рис. 3.5. Результат третьей итерации

### 3.3. Моделирование целевого сценария системы на основе графа операций

#### 3.3.1. Описание целевого сценария в виде сети Петри

Исходными данными для разработки целевого сценария процесса служит набор приоритетных целей, сформированный согласно технологии, рассмотренной в разд. 1.4, и набор операций, направленных на достижение этих целей. Обозначения и содержание операций приведены ниже.

Обозначение	Содержание операции
$f_1$	Анализ состояния системы
$f_2$	Прогнозирование развития системы и выбор стратегии

Обозначение	Содержание операции
$f_3$	Усовершенствование продукции при существующей структуре системы
$f_4$	Радикальная перестройка системы (реинжиниринг)
$f_5$	Разработка и освоение новой продукции с учетом перестройки системы
$f_6$	Формирование заказов на продукцию
$f_7$	Производство продукции
$f_8$	Сбыт (реализация) продукции

При балльной оценке значений целевых переменных  $e_1, \dots, e_4$  исходят из того, что лингвистическим значениям «низкая», «средняя», «высокая» степень достижения цели соответствуют числа 1, 2, 3. Начальные значения целевых переменных  $e_1 = 0$ ,  $e_2 = 0$ ,  $e_3 = 0$ ,  $e_4 = 0$ .

Связь между операциями целевого сценария  $f_i$ ,  $i = 1, \dots, m$ , и целевыми переменными  $e_j$ ,  $j = 1, \dots, n$ , отражена в табл. 3.2. Если при выполнении  $f_i$  изменяется значение  $e_j$ , то, как уже указывалось выше, цель  $e_j$  линейно зависит от времени операции  $f_i$  и эта зависимость выражается коэффициентом  $+\alpha_{ij}$  при возрастании  $e_j$  и коэффициентом  $-\alpha_{ij}$  при убывании  $e_j$ .

Значение коэффициента  $+\alpha_{ij}$  ( $-\alpha_{ij}$ ) задается экспертом и вписывается в табл. 3.2 на пересечении строки  $f_i$  и столбца  $e_j$  (клетку  $f_i e_j$  оставляем пустой, если при выполнении  $f_i$  значение  $e_j$  не изменяется, т.е.  $\alpha_{ij} = 0$ , либо безразлично).

Формальное описание целевого сценария в виде сети Петри для рассматриваемого примера дано на рис. 3.6. Операции  $f_i$ ,  $i = 1, \dots, m$  (позиции сети Петри) изображены прямоугольниками, переходы  $t_k$ ,  $k = 1, \dots, h$ , – жирными планками. Внутри прямоугольников (операций  $f_i$  характеризующихся изменением целевых переменных  $e_j$ ) помещены иллюстративные примеры линейных графиков в системе координат  $t_i - e_j$ . При этом одной и той же цели  $e_j$  может соответствовать несколько графиков вследствие того, что в данную операцию  $f_i$  ведет несколько переходов. Приход маркера в  $f_i$  разными маршрутами определяет и разные начальные значения  $e_j$ .

В сети Петри, представленной на рис. 3.6, такая многозначность имеет место для целевой переменной  $e_2$  в операциях  $f_6, f_7, f_8$ . В операцию  $f_6$  ведут переходы  $t_2, t_5, t_6$ , которым соответствуют линейные графики  $t_2 e_2$ ,  $t_5 e_2$ ,  $t_6 e_2$  внутри позиции  $f_6$ . Ввиду того

**Коэффициенты зависимости целевых переменных  
от продолжительности операций**

$f_i$	$e_j$			
	$e_1$	$e_2$	$e_3$	$e_4$
$f_1$	+0,5	+1,0	+0,3	+1,0
$f_2$	+0,5	+0,1	+0,6	+0,3
$f_3$		+0,2		
$f_4$	+0,65	+0,7	+0,6	+0,9
$f_5$	+0,65	+0,7	+0,6	+0,9
$f_6$		+0,1		
$f_7$		+0,1		
$f_8$		+0,1		

что эти графики порождают разные конечные значения  $e_2$ , отождествляемые с начальными значениями  $e_2$  в позиции  $f_7$ , в нее также помещены три графика для  $e_2$ . Аналогичное имеет место и для позиции  $f_8$ .

Для полноты модели операциям в сети Петри на рис. 3.6 должны также быть соотнесены алгоритмы, формирующие значения внутренних и выходных переменных моделируемого процесса. Однако для простоты изложения эти алгоритмы мы опускаем.

Рассмотрим динамику работы модели процесса на рис. 3.6. При этом исходим из того, что в сетях Петри (графах операций) переходы срабатывают мгновенно, причем в один момент только один переход, а операции «развернуты» во времени. В начальном состоянии помечена (маркирована) только операция  $f_1$ , остальные операции не помечены. Проводится анализ состояния и резервов системы, в которой реализуется процесс. По завершении анализа срабатывает переход  $t_1$ , и маркер переходит к операции  $f_2$ . Последняя по завершении своего выполнения формирует три возможные стратегические альтернативы:

- «ничего не менять» – срабатывание перехода  $t_2$  и перемещение метки из  $f_2$  в  $f_6$ ;
- усовершенствовать продукцию без коренных структурных изменений в системе – срабатывание  $t_3$  и переход маркера от  $f_2$  к  $f_3$ ;
- провести радикальную перестройку (реинжиниринг) системы – срабатывание  $t_4$ , сопровождаемое изъятием маркера из  $f_2$  и помещением по маркеру в  $f_4$  и  $f_5$ .

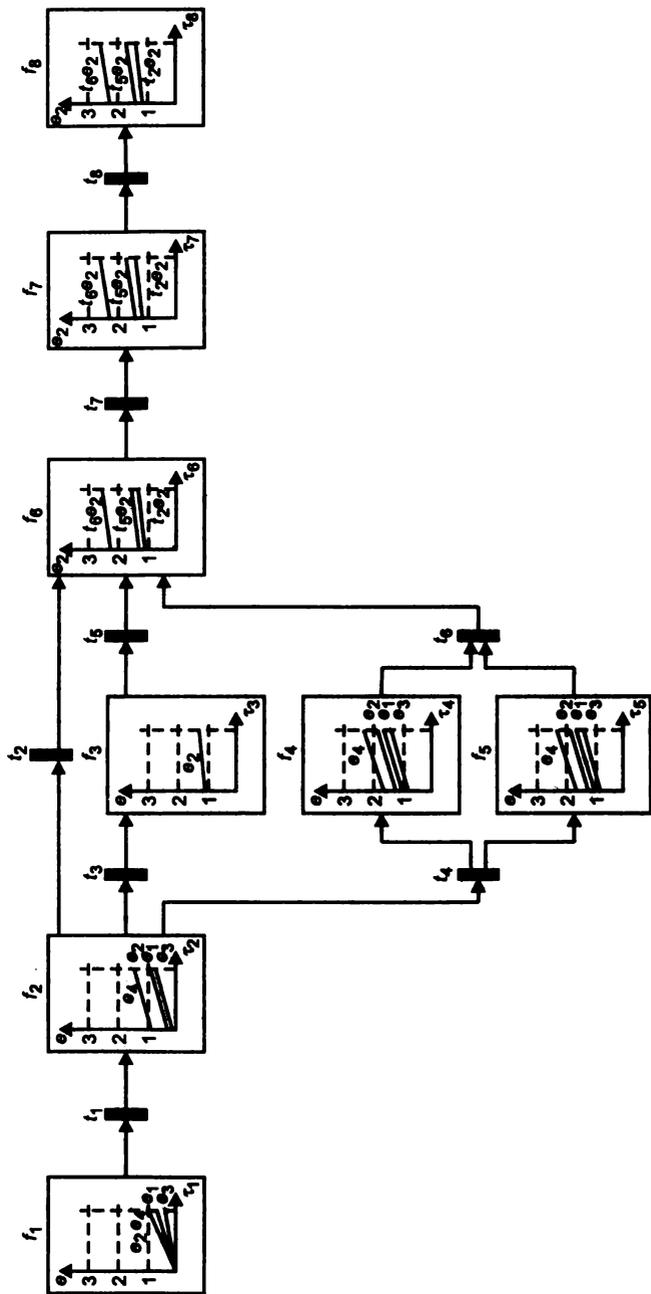


Рис. 3.6. Базовое описание целевого сценария (в виде сети Петри)

Завершение выполнения операций  $f_4$  и  $f_5$  выражается срабатыванием перехода  $t_6$ , маркеры изымаются из  $f_4$  и  $f_5$ , и помечается операция  $f_6$ . Далее маркер последовательно переходит к  $f_7$  и  $f_8$ .

### 3.3.2.

#### **Формальное описание событий, инициирующих переходы на графе операций**

Событиям будем сопоставлять булевы переменные  $s_i \in S = \{s_1, \dots, s_n\}$ , принимающие единичное значение в момент  $\tau_{s_i}$  наступления событий и нулевое значение во все остальные моменты времени. События  $s_i$ , как уже говорилось ранее, зависят от переменных из множества  $A = E \cup F \cup T \cup \Theta \cup X \cup Y \cup Z$ , причем эта зависимость определяется композицией базовых отношений на множестве событий [20].

Событие  $s_i(a_1, a_2)$ , где  $a_1, a_2 \in A$  и являются числовыми переменными, назовем элементарным сравнением, если  $a_1, a_2$  находятся в одном из отношений: «равно» ( $=$ ), «больше» ( $>$ ), «больше или равно» ( $\geq$ ). При этом одна из переменных  $a_1, a_2$  может вырождаться в константу, а в случае отношения «равно» константа может представляться лингвистическим выражением.

В общем случае событие «композируется» из переменных  $a_i$  (и элементарных сравнений) путем применения базовых «событийных» отношений.

В табл. 3.3 приводится установленный из опыта набор таких отношений [20]. В строках табл. 3.3 для отношений И, ИЛИ, НЕ показано, что множества моментов наступления событий-результатов определяются соответственно пересечением, объединением и дополнением аналогичных множеств для исходных событий,  $\{\tau\}$  – множество всех моментов на временной шкале. Отношения предшествования (следования) отображают ситуацию, когда текущий момент  $\tau$  наступает одновременно или позже (раньше) момента наступления события  $s_i$  (моменты появления событий отмечены жирными точками). Отношение «интервал» формирует отрезок временной шкалы  $\{\tau\}$ , ограниченный моментами появления событий  $s_1$  и  $s_2$ , а отношение «счет» – отрезок  $\{\tau\}$ , ограниченный моментами  $k$ -го и  $k+1$ -го появления  $s_i, k = 1, \dots, p$ . Наконец, отношение «задержка» сдвигает момент появления  $s_i$  вправо по временной оси на  $\omega$  единиц. Заметим, что набор базовых отношений (см. табл. 3.3) является открытым и может быть дополнен.

Таблица 3.3

Базовые отношения на множестве событий

Наименование отношения	Формальное описание	Временной график
Логическая связка И (конъюнкция)	$s = a_1 \wedge a_2$ $\{\tau_s\} = \{\tau_{a_1}\} \cap \{\tau_{a_2}\}$	
Логическая связка ИЛИ (дизъюнкция)	$s = a_1 \vee a_2$ $\{\tau_s\} = \{\tau_{a_1}\} \cup \{\tau_{a_2}\}$	
Логическая связка НЕ (отрицание)	$s = \bar{a}$ $\{\tau_s\} = \{\tau\} \setminus \{\tau_a\}$	
Предшествование	$Q(s_j) : \tau_{s_j} \leq \tau$	
Следование	$R(s_j) : \tau_{s_j} > \tau$	
Интервал	$I(s_1 / s_2) :$ $(\tau_{s_1} \leq \tau) \wedge (\tau < \tau_{s_2}) =$ $= Q(s_1) \wedge R(s_2)$	
Счет	$N^k(s_j) :$ $(\tau_{s_j}^k \leq \tau) \wedge (\tau < \tau_{s_j}^{k+1})$	
Задержка	$D^\omega(s_j) :$ $\tau_D = \tau_{s_j} + \omega$	

Формальное описание на основе введенных правил событий, инициирующих переходы на графе операций (см. рис. 3.6), приведено ниже.

Обозначение	Формула события
$s(t_1)$	$((e_1 \geq 0,5) \wedge (e_2 \geq 1) \wedge (e_3 \geq 0,3) \wedge (e_4 \geq 1)) \vee \vee D^{\omega_1}(f_1)) \wedge \bar{I}(x_1 / x_2)$
$s(t_2)$	$((e_1 \geq 1) \wedge (e_2 \geq 1,2) \wedge (e_3 \geq 0,6) \wedge (e_4 \geq 1,5)) \vee \vee D^{\omega_2}(f_2)) \wedge \bar{I}(x_1 / x_2) \wedge (z_1 = \text{стратегия1})$
$s(t_3)$	$((e_1 \geq 1) \wedge (e_2 \geq 1,2) \wedge (e_3 \geq 0,6) \wedge (e_4 \geq 1,5)) \vee \vee D^{\omega_2}(f_2)) \wedge \bar{I}(x_1 / x_2) \wedge (z_1 = \text{стратегия2})$
$s(t_4)$	$((e_1 \geq 1) \wedge (e_2 \geq 1,2) \wedge (e_3 \geq 0,6) \wedge (e_4 \geq 1,5)) \vee \vee D^{\omega_2}(f_2)) \wedge \bar{I}(x_1 / x_2) \wedge (z_1 = \text{стратегия3})$
$s(t_5)$	$((e_2 \geq 1,3) \vee D^{\omega_3}(f_3)) \wedge \bar{I}(x_1 / x_2)$
$s(t_6)$	$((\min(f_4 e_1, f_5 e_1) \geq 1,8) \wedge (\min(f_4 e_2, f_5 e_2) \geq 2,0) \wedge (\min(f_4 e_3, f_5 e_3) \geq 1,9) \wedge (\min(f_4 e_4, f_5 e_4) \geq 2,5)) \vee \vee D^{\omega}(f_4 \wedge f_5)) \wedge \bar{I}(x_1 / x_2), \text{ где } \omega = \max(\omega_4, \omega_5)$
$s(t_7)$	$((e_2 \geq 2,7) \wedge Q(t_6)) \vee ((e_2 \geq 1,9) \wedge Q(t_5)) \vee ((e_2 \geq 1,5) \wedge Q(t_2)) \vee \vee D^{\omega_6}(f_6)) \wedge \bar{I}(x_1 / x_2)$
$s(t_8)$	$((e_2 \geq 2,8) \wedge Q(t_6)) \vee ((e_2 \geq 2,0) \wedge Q(t_5)) \vee ((e_2 \geq 1,6) \wedge Q(t_2)) \vee \vee D^{\omega_7}(f_7)) \wedge \bar{I}(x_1 / x_2)$

Событие  $s(t_1)$  наступает при достижении линейно возрастающими при выполнении операции  $f_1$  целевыми переменными  $e_1, \dots, e_4$  указанных значений (выражается через элементарные сравнения) или по истечении нормативного времени  $\omega_1$  операции  $f_1$  при обязательном условии отсутствия блокирующего события  $I(x_1/x_2)$ , инициируемого при значении входной переменной  $x_1=1$  и отменяемого при  $x_2=1$ .

События  $s(t_2), s(t_3), s(t_4)$  имеют аналогичную структуру, но при более высоких значениях  $e_1, \dots, e_4$ , достигаемых во время операции  $f_2$ , и дополнительно характеризуются выходной лингвистической переменной  $z_1$ , отображающей выбранную стратегию раз-

вития системы. Событие  $s(t_3)$ , определяющее завершение операции  $f_3$ , зависит только от целевой переменной  $e_2$  (на остальные цели  $f_3$  не влияет).

Событие  $s(t_6)$  фиксирует завершение параллельно выполняемых операций  $f_4, f_5$ , влияющих на изменение всех четырех целевых переменных. Поскольку цели  $e_i, i=1, \dots, 4$ , в этих операциях принимают разные конечные значения  $f_4 e_i, f_5 e_i$ , то в описание события введены элементарные сравнения по минимальному значению –  $\min(f_4 e_i, f_5 e_i)$ , а время задержки  $\omega$  определено максимальным из нормативных времен  $\omega_4, \omega_5$ .

События  $s(t_7), s(t_8)$ , подобно  $s(t_5)$ , зависят только от целевой переменной  $e_2$ , однако достигаемые в операциях  $f_6, f_7$  конечные значения  $e_2$  определяются предысторией – какой из альтернативных переходов  $t_2, t_5, t_6$  инициировал  $f_6$ . Для учета предыстории в выражения событий вводим члены  $Q(t_6), Q(t_5), Q(t_2)$ , отражающие отношение предшествования (см. табл. 3.3).

### 3.3.3.

#### Равносильные преобразования описания событий

Над формулами событий, примеры которых были даны выше, можно выполнять равносильные преобразования, приводящие к упрощению моделей функционирования процессов. Остановимся на этом вопросе подробнее.

«Событийные» отношения, представленные в табл. 3.3, будем интерпретировать как операторы, отображающие булевы переменные  $s_i, i = 1, \dots, l$ , входных событий в булеву переменную  $s$  выходного (результатирующего) события. Операторная формула [20] – это выражение, полученное путем применения конечного числа раз к переменным  $s_i$  операторов  $\wedge, \vee, -, Q, R, I, N, D$  согласно табл. 3.3.

Приведем пример операторной формулы

$$s = (s_1 \vee I(s_1 / s_2) \wedge I(s_1 / s_3)) \wedge (s_2 \vee s_3). \quad (3.2)$$

Операторная формула представима в виде операторной схемы, на которой операторы изображаются прямоугольниками, а переменные – стрелками. Операторы разбиты на каскады: к первому каскаду относят операторы, примененные только по входным переменным, ко второму каскаду – операторы, примененные к операторам первого каскада (и, возможно, ко входным

переменным) и т.д. Формуле (3.2) соответствует четырехкаскадная схема на рис. 3.7 (включающий и выключающий входы оператора  $I$  помечены соответственно знаками «+», «-»).

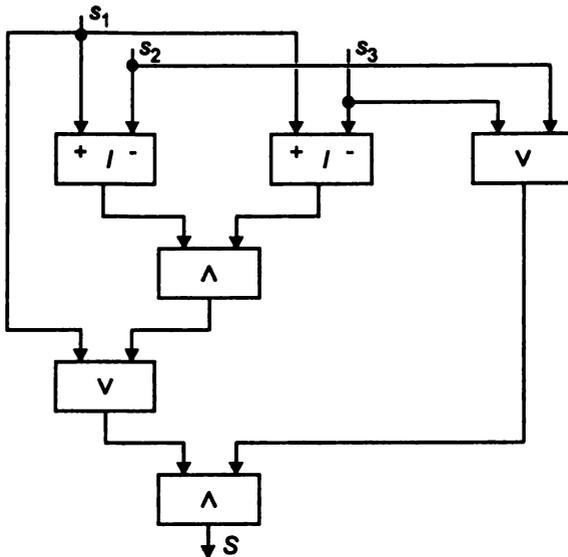


Рис. 3.7. Пример операторной схемы

Равносильные преобразования формул на базе операторов И, ИЛИ, НЕ ( $\wedge$ ,  $\vee$ ,  $\neg$ ) определяются известными правилами алгебры логики (булевой алгебры), приведенными в табл. 3.4.

Для остальных операторов, приведенных в табл. 3.3, также имеет место определенный набор правил [20]. Не претендуя (в рамках данного учебного пособия) на глубокое обсуждение проблемы равносильных преобразований операторных формул, а преследуя лишь цель иллюстрации идеи механизма преобразования, дадим в качестве примера правило для оператора *Интервал*  $I(s_1/s_2)$ :

1.  $I(s_1/\bar{s}_1) = s_1$ .
2.  $s_2 \wedge I(s_1/s_2) = 0$ .
3.  $s_1 \wedge I(s_1/s_2) = s_1$ .

4.  $s_1 \vee I(s_1 / s_2) = I(s_1 / s_2)$ .
5.  $s_2 \vee I(s_1 / s_2) = I(s_1 / \overline{s_2})$ .
6.  $I(s_1 \vee s_2 / s_3) = I(s_1 / s_3) \vee I(s_2 / s_3)$ .
7.  $I(s_1 / s_2 \vee s_3) = I(s_1 / s_2) \wedge I(s_1 / s_3)$ .
8.  $I(s_1 / s_2) = \overline{I(s_2 / s_1)}$ .
9.  $I(1/0) = 1$ .
10.  $I(0/1) = 0$ .

Таблица 3.4

Правила алгебры логики

№ п/п	Название правила	Символическая запись
1	Коммутативность	$s_1 \vee s_2 = s_2 \vee s_1; s_1 \wedge s_2 = s_2 \wedge s_1$
2	Ассоциативность	$(s_1 \vee s_2) \vee s_3 = s_1 \vee (s_2 \vee s_3);$ $(s_1 \wedge s_2) \wedge s_3 = s_1 \wedge (s_2 \wedge s_3)$
3	Дистрибутивность	$s_1 \wedge (s_2 \vee s_3) = (s_1 \wedge s_2) \vee (s_1 \wedge s_3);$ $s_1 \vee (s_2 \wedge s_3) = (s_1 \vee s_2) \wedge (s_1 \vee s_3)$
4	Поглощение	$s_1 \vee (s_1 \wedge s_2) = s_1; s_1 \wedge (s_1 \vee s_2) = s_1$
5	Тавтология	$s_1 \vee s_1 = s_1; s_1 \wedge s_1 = s_1$
6	Соотношения с константами	$s_1 \vee 1 = 1; s_1 \wedge 1 = s_1; s_1 \vee 0 = s_1;$ $s_1 \wedge 0 = 0; \overline{0} = 1; \overline{1} = 0$
7	Дополнительность	$s_1 \vee \overline{s_1} = 1; s_1 \wedge \overline{s_1} = 0$
8	Распространение	$(s_1 \wedge s_2) \vee (s_1 \wedge \overline{s_2}) = s_1;$ $(s_1 \vee s_2) \wedge (s_1 \vee \overline{s_2}) = s_1$
9	Двойственность (правило де-Моргана)	$\overline{s_1 \vee s_2} = \overline{s_1} \wedge \overline{s_2}; \overline{s_1 \wedge s_2} = \overline{s_1} \vee \overline{s_2}$
10	Удвоенное отрицание	$\overline{\overline{s_1}} = s_1$
11	Противоположность	Если $s_1 = \overline{s_2}$ , то $\overline{s_1} = s_2$

На основе приведенных правил преобразуем формулу (3.2). С учетом правила 7 (см. правила для оператора *Интервал*) получаем:

$$s = (s_1 \vee I(s_1 / s_2 \vee s_3)) \wedge (s_2 \wedge s_3). \quad (3.3)$$

К (3.3) применяем правило 4.

$$s = I(s_1/s_2 \vee s_3) \wedge (s_2 \vee s_3). \quad (3.4)$$

Далее согласно правилу 2 формула (3.3) выражает нулевую константу

$$s = 0. \quad (3.5)$$

### 3.3.4.

#### **Моделирование функционирования процессов при нештатных ситуациях**

В ходе выполнения процесса могут возникать «нештатные» ситуации (например, превышение установленного предела брака выпускаемой продукции, недопустимое падение запаса ресурсов и т.д.), требующие изменения порядка его функционирования.

При нештатных ситуациях выполнение процесса прерывается и проводятся необходимые действия по идентификации ситуации и устранению ее причин. Прерывание может осуществляться двумя способами: останов процесса в текущем состоянии либо перевод его в некоторое предыдущее состояние, чаще всего начальное. Последний способ прерывания реализуется в результате выполнения процесса «против течения» – с помощью так называемого обратного срабатывания переходов (изъятия маркеров из выходных и внесения во входные позиции перехода).

Для моделирования прерывания путем останова процесса в операторные формулы условий переходов графа операций вводятся специальные подформулы, которые разрешают срабатывание переходов при «штатном» выполнении процесса и блокируют их при «нештатных» ситуациях. В формулах, приведенных выше, это подформула  $\bar{I}(x_1 / x_2)$ , где  $x_1, x_2$  – входные переменные, фиксирующие соответственно появление и устранение «нештатных» ситуаций (при  $x_1 = 1, x_2 = 0, I(x_1 / x_2) = 1, \bar{I}(x_1 / x_2) = 0, s(t_i) = 0, i = 1, \dots, 8$ ).



2. Все выделенные формулы «запускаются», и осуществляются указанные в них присваивания (для предотвращения неопределенности параллельно запускаться могут лишь такие формулы, которые не присваивают разных значений одной и той же переменной).

3. Повторяются шаги 1 и 2 до тех пор, пока не установится равновесие – условия всех формул не будут выполняться.

Пусть, например, в момент появления «нештатной» ситуации  $I(x_1/x_2)=1$  граф операций (см. рис. 3.6) находился в маркировке  $f_6=1$ , в которую он был приведен срабатыванием перехода  $t_5$ . При этом запускается формула в строке  $t_5$  (см. с. 74) и устанавливается маркировка  $f_3 = 1$ , в которой запускается формула в строке  $t_3$ , приводящая к маркировке  $f_2 = 1$ . Далее запускается формула в строке  $t_1$  и устанавливается равновесие (в маркировке  $f_1 = 1$ ).

### 3.3.5.

#### **Имитационное моделирование целевого сценария на основе графа операций**

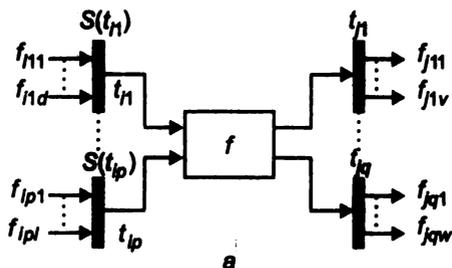
Имитационное моделирование целевого сценария осуществляется путем «прогона» компьютерной программы, реализующей граф операций. Парадигма программирования на основе графа операций заключается в следующем.

Граф представляется в виде композиции однотипных стандартных блоков по числу операций. Эти блоки будем называть нормализованными бинарными компонентами, а процедуру преобразования графа операций – нормализацией. Любая бинарная компонента может быть преобразована в программный модуль путем настройки (адаптации) универсального программного «шаблона», написанного на высокоуровневом языке программирования (например, C++, Delphi и др.). Сформированные таким образом программные модули для операций компилируются в общую программу с учетом связей между бинарными компонентами, указанными в нормализованном описании графа операций.

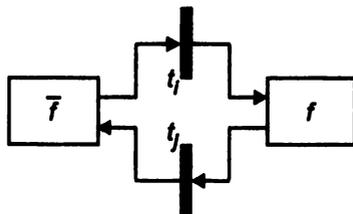
Процедура нормализации поясняется рис. 3.8. На рис. 3.8, *a* показан типовой фрагмент графа операций, содержащий позицию операции  $f$  и ее входные (выходные) переходы  $t_{i1}, \dots, t_{ip}$  ( $t_{j1}, \dots, t_{jq}$ ), а также обозначение операций-предшественников (последователей) относительно  $f$ .

На рис. 3.8, *б* изображена нормализованная бинарная компонента графа, сопоставленная операции  $f$ . Она содержит две пози-

ции, одна из которых ( $f$ ) соответствует выполнению этой операции, а другая ( $\bar{f}$ ) – ее невыполнению, т.е. реализации любой из остальных операций. Переход  $t_i$ , ведущий из  $\bar{f}$  в  $f$ , инициируется событием  $s(t_i)$ , представляющим собой дизъюнкцию конъюнкций, образованных событием для входного перехода и операционными переменными для всех его предшественников. Переход  $t_j$ , ведущий из  $f$  в  $\bar{f}$ , инициируется событием  $s(t_j)$  – дизъюнкцией (по всем выходным переходам  $f$ ) конъюнкций операционных переменных всех последователей.



$$s(t_i) = (s(t_{i1}) \wedge f_{i11} \wedge \dots \wedge f_{i1d}) \vee \dots \vee (s(t_{ip}) \wedge f_{ip1} \wedge \dots \wedge f_{ipl})$$



$$s(t_j) = (f_{j11} \wedge \dots \wedge f_{j1v}) \vee \dots \vee (f_{jq1} \wedge \dots \wedge f_{jqw})$$

Рис. 3.8. Нормализация графа операций:

*a* – типовой фрагмент графа операций;

*б* – нормализованная бинарная компонента графа операций

Результат нормализации графа операций (см. рис. 3.6) показан на рис. 3.9. Начальная маркировка графа операций соответствует

помещению маркеров в позиции  $f_1, \bar{f}_2, \dots, \bar{f}_8$ , причем в  $f_1$  маркер был переведен из  $\bar{f}_1$  внешним (пусковым) событием  $x_3 = 1$ . При наступлении события  $s(t_1)$  при условии  $f_1 = 1$  маркер переходит из  $\bar{f}_2$  в позицию  $f_2$ , а это, в свою очередь, вызывает переход маркера из  $f_1$  в  $\bar{f}_1$ . Таким образом, имеет место контролируемый двухшаговый процесс взаимодействия бинарных компонент графа операций.

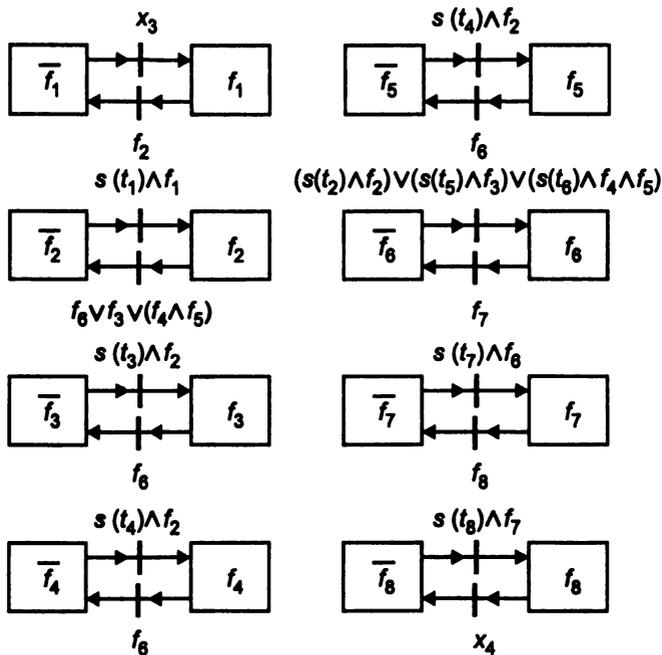


Рис. 3.9. Результат нормализации графа операций, приведенного на рис.3.6

В позиции  $f_2$  маркер будет находиться до тех пор, пока не наступит событие  $(f_6 = 1) \vee (f_3 = 1) \vee ((f_4 = 1) \wedge (f_5 = 1))$ , возвращающее маркер в  $\bar{f}_2$ , и т.д. Из позиции  $f_8$  маркер передается в  $\bar{f}_8$  внешним («сбрасывающим») событием  $x_4=1$ .

Как нетрудно убедиться, граф операций (см. рис. 3.6) и многокомпонентная нормализованная структура (см. рис. 3.9) «ведут себя одинаково» (эквивалентны по поведению).

При программировании по графу операций наряду с настраиваемыми шаблонами для нормализованных бинарных компонент графа используются шаблоны элементарных событий и правила композиции из них «событийных» программ.

Кратко обсудим форму представления результатов имитационного моделирования целевого сценария – графа операций. Такими результатами, как уже говорилось, являются графики изменения на единой системной временной шкале целевых, внутренних и выходных переменных процесса.

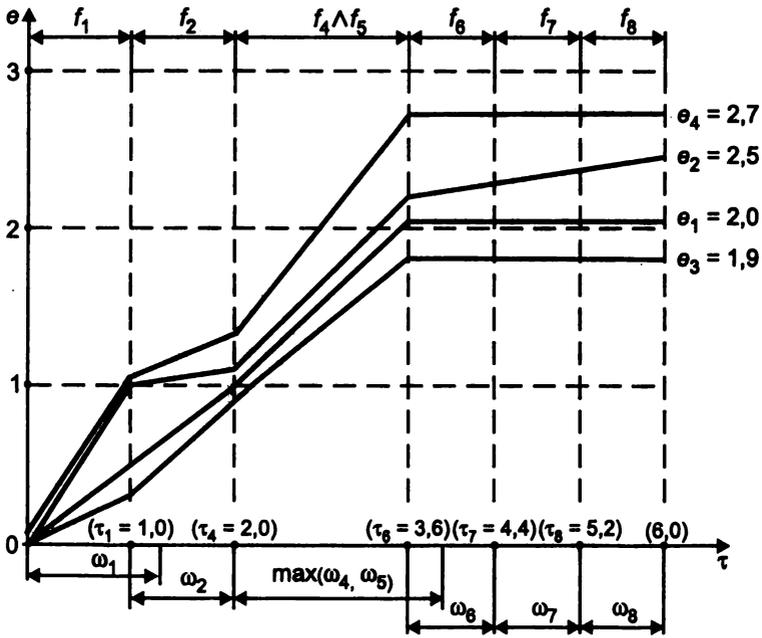


Рис. 3.10. Результат имитационного моделирования целевого сценария – временные графики целевых переменных

На рис. 3.10 даны графики целевых переменных  $e_1, \dots, e_4$ , полученные по графу операций, заданному сетью Петри на рис. 3.6 и таблицей событий переходов (см. разд. 3.3.2).

Имитационный эксперимент проводился при значениях параметров операций, указанных в табл. 3.2, и соответствует последовательности срабатывания переходов  $t_1, t_4, t_6, t_7, t_8$ . Через  $\tau_i$  на рис. 3.10 обозначены моменты срабатывания переходов  $t_i$ .

Момент  $\tau_1$  перехода от операции  $f_1$  к  $f_2$  соответствует достижению всеми целями уровней, указанных в строке  $s(t_1)$  (см. с. 69), и наступает раньше истечения нормативного (планового) времени  $\omega_1$ . Эксперимент дает фактическую длительность операции  $f_1$  (и ее отклонение от плановой). Момент  $\tau_4$  перехода от  $f_2$  к выполнению  $f_4$  и  $f_5$  определяется истечением нормативного времени  $\omega_2$  согласно строке  $s(t_4)$  (см. с. 69). При этом по графикам (см. рис. 3.10) определяются уровни достижения всех целей в момент  $\tau_4$ .

Параллельные операции  $f_4, f_5$  одновременно воздействуют на все четыре целевые переменные. Если параллельные операции, влияющие на общую цель  $e_i$ , имеют разные параметры, т.е. по-разному воздействуют на  $e_i$  (одна операция может приближать к цели, а другая удалять от нее и т.д.), то график изменения  $e_i$  на интервале параллельных операций формируется по средним значениям. В нашем примере (см. табл. 3.2) параметры  $f_4$  и  $f_5$  совпадают и график строится по любой из них. Момент  $\tau_6$  перехода от выполнения параллельных операций  $f_4, f_5$  наступает до истечения нормативного времени  $\omega = \max(\omega_4, \omega_5)$  при достижении целями  $e_1, e_2, e_3, e_4$  соответственно уровней 2,0; 2,2; 1,9; 2,7. Фактическая длительность более продолжительной из операций  $f_4, f_5$  определяется по графику. На интервалах операций  $f_6, \dots, f_8$  изменяется только значение целевой переменной  $e_2$ , переходы происходят по истечении нормативного времени.

Проведенный имитационный эксперимент над целевым сценарием показывает, что при принятых величинах параметров операций и «уставок» в выражениях событий на графе операций целевые переменные  $e_1, e_2, e_3, e_4$  принимают конечные значения 2,0; 2,5; 1,9; 2,7, которые можно сравнить с рекомендуемыми плановыми значениями 2,0; 3,0; 2,0; 3,0 (установленными на основе технологии целеполагания). Решение задачи – удовлетвориться фактическими уровнями достижения целей либо провести коррекцию графа операций и повторить эксперимент, возлагается на лицо, принимающее решение (ЛПР).

### 3.4. Моделирование сценария логического управления распределенных систем на основе взаимодействующих графов операций

#### 3.4.1. Временные взаимодействия графов операций

Временные взаимодействия между графами операций процессов заключаются в том, что срабатывание некоторого перехода в графе инициируется одновременным наступлением двух событий: собственного события, влияющего на функционирование данного графа, и разрешающего события, отражающего развитие ситуации на графе другого процесса. Формулы собственного и разрешающего событий соединяются знаком логической связки И (конъюнкции).

Элементарные временные взаимодействия графов операций, отражающие базовые отношения на множестве событий, проиллюстрированы в табл. 3.5 (на примере фрагментов графов с операциями и переходами соответственно  $f, t; p, q$ ).

Дадим краткие пояснения к табл. 3.5.

1. Синхронизация срабатывания переходов ( $t$  и  $q$ ) в разных графах операций означает, что  $t$  и  $q$  срабатывают одновременно и только одновременно. Они инициируются общим событием  $S$ . При этом тот граф операций, который первым приходит во входные позиции синхронизированного перехода, ожидает в этом состоянии, пока другой граф не придет в аналогичные позиции (из-за этого возможны потери производительности). Здесь и далее  $p, q_i$  интерпретируются как логические (булевы) переменные.

2. Блокировка перехода  $t$  операцией  $q$  другого графа разрешает срабатывание  $t$  лишь во время выполнения  $q$ , т.е. при  $q = 1$ .

3. Блокировка перехода по предыстории другого графа операций разрешает срабатывание  $t$  лишь при условии, что ранее (либо одновременно) этот другой граф выполнил (выполняет) операцию  $q$ , т.е. имеет место  $Q(q) = 1$ .

4. Блокировка перехода по постистории другого графа операций аналогична предыдущей с тем отличием, что другой граф еще не приступал к выполнению операции  $q$ :  $R(q) = 1$ .

Элементарные временные взаимодействия графов операций

№ п/п	Тип отношения	Пример
1	Синхронизация переходов в разных графах ( $S = S(t) = S(q)$ )	<p><math>S \wedge p_1 \wedge p_2</math>      <math>S \wedge f_1 \wedge f_2</math></p>
2	Блокировка по текущей операции в другом графе	<p><math>S(t) \wedge q</math></p>
3	Блокировка по предыстории другого графа	<p><math>S(t) \wedge Q(q)</math></p>
4	Блокировка по постистории другого графа	<p><math>S(t) \wedge R(q)</math></p>
5	Блокировка по межоперационному интервалу на другом графе	<p><math>S(t) \wedge I(q_i / q_j)</math></p>
6	Взаимодействие через "почтовый ящик"	

5. Блокировка перехода по интервалу на другом графе, начинающемся операцией  $q_i$  и заканчивающемся в момент инициирования операции  $q_j$  ( $I(q_i/q_j) = 1$ ), подразумевает разрешение на срабатывание  $t$  лишь в моменты, принадлежащие этому интервалу.

6. Каждое срабатывание перехода  $t$  фиксируется в почтовом ящике (Box). Переход  $q$  может сработать столько раз (или меньше), сколько сработал переход  $t$ .

Разумеется, временные взаимодействия графов операций могут определяться более сложными событиями, представляющими собой композицию элементарных.

### 3.4.2.

#### Обменные взаимодействия графов операций

В качестве примера обменных взаимодействий при децентрализованной и централизованной архитектуре системы рассмотрим механообрабатывающий цех-автомат (рис. 3.11), представляющий собой нижний уровень рассматриваемой производственной организационной системы.

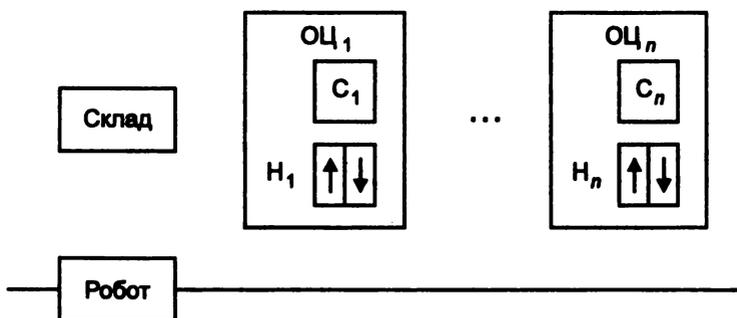


Рис. 3.11. Схема механообрабатывающего цеха-автомата

В его состав входят исполнительные структурные компоненты: обрабатывающие центры (ОЦ), транспортный робот и автоматизированный склад. ОЦ – это автоматический комплекс с числовым программным управлением, предназначенный для определенного вида механообработки (токарной, фрезерной и т.д.), включающий станок С (оснащенный инструментальным магази-

ном), накопитель  $H$  для заготовок и обработанных деталей, перегрузочные манипуляторы. На складе размещены заготовки, инструмент, оснастка для всех ОЦ,  $i=1, \dots, n$ , на планируемый период времени, обработанные детали со всех ОЦ, комплексы программ и т.д. Робот по мере необходимости пополняет накопители  $H_i$  заготовками со склада и отвозит из накопителей на склад обработанные детали. Станки  $C_i$  периодически переналаживаются на обработку запланированной партии деталей со сходными технологическими параметрами (принцип групповой обработки). Исходная позиция робота находится у склада, характеристика его транспортного цикла (маршрут и перевозимый груз) определяется плановыми заданиями и фактической ситуацией в цехе.

Диаграмма обменных взаимодействий между вышерассмотренными компонентами (децентрализованный вариант) приведена на рис. 3.12.

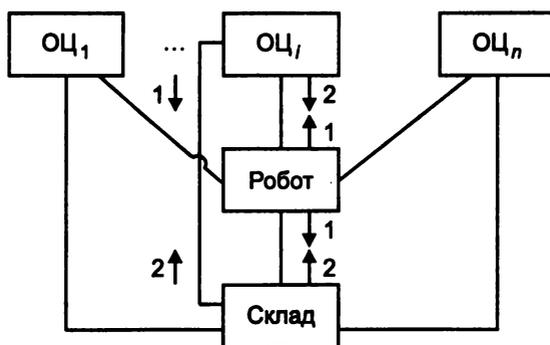


Рис. 3.12. Диаграмма обменных взаимодействий по децентрализованному варианту

Она имеет вид неориентированного графа, вершины которого сопоставлены компонентам, ребра – факту взаимодействия между компонентами, а рядом с ребрами проставлены пронумерованные стрелки, соответствующие передаваемым сообщениям (номера стрелок указывают порядок передачи сообщений по данному ребру). Содержание сообщений для диаграммы на рис. 3.12 приведено в табл.3.6.

Пояснения к диаграмме, изображенной на рис. 3.12

Взаимодействующие компоненты	Номер сообщения	Содержание сообщения
Робот – ОЦ <sub>i</sub>	1	Запрос о состоянии накопителя Н <sub>i</sub>
	2	Данные о текущем состоянии Н <sub>i</sub>
Робот – склад	1	Запрос о запасе заготовок на складе
	2	Данные о запасе заготовок на складе
ОЦ <sub>i</sub> – склад	1	Запрос об обеспеченности заготовками групповой обработки
	2	Данные об обеспеченности заготовками

На рис. 3.13 даны примеры графов операций компонентов системы – робота и обрабатывающего центра ОЦ<sub>i</sub> при децентрализованном управлении.

Начнем с графа операций робота (см. рис. 3.13, а).

Операция  $f_1$  принятия решений на основе мониторинга состояния накопителей всех ОЦ и с учетом ситуации на складе определяет:

- тип транспортного цикла: *тип 1* – загрузка ОЦ<sub>i</sub> с последующей разгрузкой ОЦ<sub>j</sub>, *тип 2* – только загрузка ОЦ<sub>j</sub>, *тип 3* – только разгрузка ОЦ<sub>j</sub>;
- число заготовок, транспортируемых со склада в загружаемый накопитель, и число обработанных деталей, доставляемых на склад из разгружаемого накопителя.

В начальном состоянии графа робота маркер находится в позиции операции  $f_1$ . При наступлении события  $s_1=1$ , означающего, что решение принято, срабатывает переход  $t_1$  и маркер передается от  $f_1$  к операции  $f_2$ . При циклах типа 1 и 2 робот забирает со склада вычисленное число заготовок и движется к ОЦ<sub>j</sub>, выбранному для загрузки. При цикле типа 3 робот не берет заготовок и «пустой» движется к разгружаемому ОЦ<sub>j</sub>. Событие  $s_2 = 1$  фиксирует приход робота к ОЦ<sub>i</sub> в рамках циклов типа 1 и 2, событие  $s_2 = 0$  ( $\bar{s}_2 = 1$ ) – в рамках цикла типа 3. В зависимости от значения  $s_2$  срабатывает переход  $t_2$  или  $t_3$ , маркер от  $f_2$  передается операции  $f_3$  (загрузка накопителя Н<sub>i</sub>) или операции  $f_4$  (разгрузка Н<sub>i</sub>).

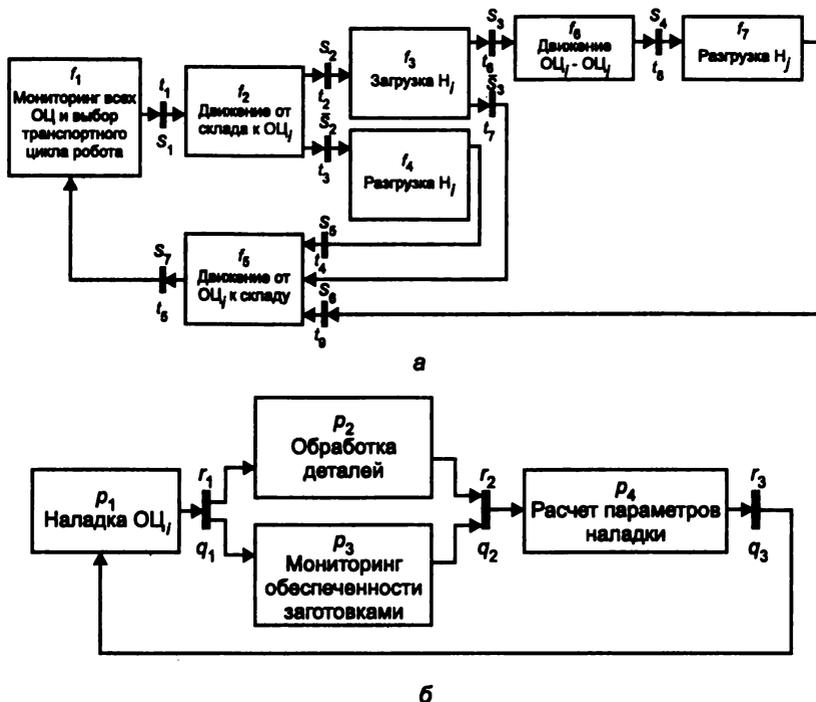


Рис. 3.13. Графы операций компонентов при децентрализованном управлении системой:  
 а – граф операций робота; б – граф операций обрабатывающего центра

Завершение загрузки  $H_i$  при выполнении операции  $f_3$  в рамках цикла типа 1 фиксируется событием  $s_3 = 1$ , вызывающим срабатывание перехода  $t_6$ , а в рамках цикла типа 2 – событием  $s_3 = 0$  ( $\bar{s}_3 = 1$ ), инициирующим переход  $t_7$ .

В цикле 1 маркер передается от операции  $f_3$  к  $f_6$  и робот движется от ОЦ<sub>*i*</sub> к ОЦ<sub>*j*</sub>, если  $i \neq j$ , либо продолжает оставаться возле ОЦ<sub>*i*</sub>, если  $i = j$ . Приход робота в позицию ОЦ<sub>*j*</sub> либо истечение заданной выдержки времени в позиции ОЦ<sub>*i*</sub> фиксируется событием  $s_4 = 1$ , по которому срабатывает переход  $t_8$ , маркер передается от операции  $f_6$  к  $f_7$  и происходит разгрузка  $H_j$ . По событию  $s_6 = 1$  (накопитель  $H_j$  разгружен) срабатывает переход  $t_9$ , маркер от  $f_7$  передается операции  $f_5$ , робот доставляет груз на склад (событие  $s_7 = 1$ , вызывающее срабатывание перехода  $t_5$  и возврат маркера операции  $f_1$ ).

При реализации цикла 2 срабатывание  $t_7$  передает маркер от  $f_3$  операции  $f_5$ , «пустой» робот возвращается на исходную позицию возле склада. При цикле 3 завершение разгрузки  $H_i$  в операции  $f_4$  отмечается событием  $s_5 = 1$  и срабатыванием перехода  $t_4$ . Цикл 3 завершается так же, как и цикл 1.

На графе операций обрабатывающего центра (см. рис. 3.13, б) операция  $p_1$  соответствует наладке ОЦ, на групповую обработку,  $p_2$  – групповой обработке деталей,  $p_3$  – мониторингу наличия на складе необходимых заготовок и принятию решения о переналадке на другой вид групповой обработки,  $p_4$  – расчету параметров этой переналадки. Операции  $p_2$  и  $p_3$  выполняются параллельно. События  $r_1 = 1, \dots, r_3 = 1$ , инициирующие срабатывание переходов  $q_1, \dots, q_3$ , означают соответственно окончание наладки ОЦ, завершение групповой обработки и принятие решения о переналадке, выбор значений параметров переналадки. В начальном состоянии помечена операция  $p_1$ , что соответствует первичной наладке ОЦ, на моделируемом временном интервале.

Перейдем к рассмотрению централизованного варианта обменных взаимодействий между компонентами цеха (рис. 3.14).

Компоненту-координатору (центру управления) передаются функции мониторинга состояния склада и накопителей в составе ОЦ, выбора транспортного цикла робота, расчета параметров переналадки ОЦ, формирования команд на запуск робота и переналадки ОЦ. Содержание сообщений, передаваемых между координатором и исполнителями, дано в табл. 3.7.

На рис. 3.15 даны примеры графов операций компонентов при централизованном управлении цехом – граф координатора (с операциями  $k_i$  и переходами  $l_i$ ) и граф ОЦ, (операции и переходы обозначены так же, как и на рис. 3.13).



Рис. 3.14. Диаграмма обменных взаимодействий по централизованному варианту

Пояснения к диаграмме, изображенной на рис. 3.14

Взаимодействующие компоненты	Номер сообщения	Содержание сообщения
Координатор – ОЦ <sub>l</sub>	1	Запрос о состоянии накопителя Н <sub>l</sub>
	2	Данные о текущем состоянии Н <sub>l</sub>
	3	Команда на переналадку ОЦ <sub>l</sub> (с указанием параметров переналадки)
Координатор – робот	1	Команда на запуск транспортного цикла робота (с указанием его типа и данных о маршруте и грузе)
Координатор – склад	1	Запрос о наличии на складе заготовок, инструмента, оснастки и т.п. (по каждому ОЦ <sub>l</sub> )
	2	Данные о текущем запасе на складе

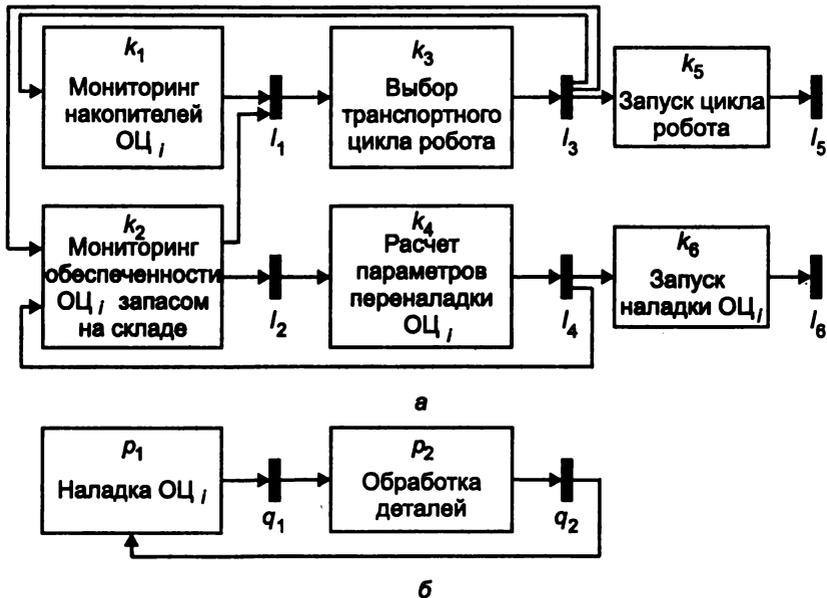


Рис. 3.15. Графы операций компонентов при централизованном управлении системой:  
 а – граф операций координатора;  
 б – граф операций обрабатывающего центра (ОЦ)

Обозначения событий, инициирующих переходы, для простоты опущены. Что касается графа операций робота, то он совпадает с приведенным на рис. 3.13, а с тем отличием, что  $f_1$  – это не операция принятия решения, а операция ожидания роботом в позиции у склада команды на выполнение транспортного цикла.

Как видно из сопоставления рис. 3.13 и рис. 3.15, при преобразовании децентрализованной архитектуры системы в централизованную графы операций исполнителей упрощаются (в графе ОЦ, на рис. 3.15 отсутствует параллелизм).

### Контрольные вопросы и упражнения

1. Какой смысл вкладывается в понятие «сценарий функционирования организационной системы»? Чем отличается целевой сценарий от сценария логического управления? Какие типы переменных и какой математический аппарат используются при описании сценариев? Сформулируйте основные задачи операционного моделирования организационных систем.
2. Дайте определение сети Петри, ее структуры и поведения. На «учебном» примере проиллюстрируйте механизм функционирования сети Петри. Сформулируйте свойства ограниченности, безопасности и «живости» сети Петри. Прокомментируйте метод анализа свойств сети Петри на основе ее ленты достижимости. Разъясните метод редукции сети Петри.
3. Охарактеризуйте способ формального описания целевых сценариев организационных систем на базе графов операций, сочетающих в себе языковые средства сетей Петри и алгебры событий. Как интерпретируются и как изображаются позиции и переходы сети Петри, составляющей основу целевого сценария? Составьте «учебный» пример такой сети и проиллюстрируйте ее динамику.
4. Укажите набор элементарных событий, инициирующих переходы на графах операций. Как из элементарных событий компонируются более сложные события? Дайте примеры равносильных преобразований описания событий.
5. Охарактеризуйте два способа прерывания нормального («штатного») функционирования организационной системы при возникновении «нештатных» ситуаций – останов процесса в текущем состоянии либо его возврат в какое-нибудь из предыдущих состояний срабатыванием переходов графа операций «против течения». Как указанные прерывания отображаются в описании целевого сценария?

6. Изложите принцип программирования на основе целевого сценария в виде графа операций. Приведите типовую схему нормализованной бинарной компоненты графа, реализуемую программным модулем. В чем заключается процедура нормализации графа операций? В какой последовательности выполняется имитационное компьютерное моделирование версий целевого сценария организационной системы? Как графически представляются результаты имитационных компьютерных экспериментов?
7. В чем заключается операционное моделирование сценариев логического управления распределенных организационных систем на основе взаимодействующих графов операций? Охарактеризуйте временные и обменные взаимодействия между графами операций (между процессами) и дайте примеры таких взаимодействий. Приведите схему обменных взаимодействий при децентрализованной и централизованной архитектуре системы. Какие вы видите преимущества и недостатки этих двух видов архитектуры организационных систем?

## Глава 4

---

# ПОТОКОВОЕ МОДЕЛИРОВАНИЕ ОРГАНИЗАЦИОННЫХ СИСТЕМ

### 4.1.

#### Принципы потокового моделирования

Процессы (операции), порядок следования которых исследовался на этапе операционного моделирования (гл. 3), взаимодействуют между собой и с внешним окружением системы посредством потоков (информационных, материальных, финансовых и др.). Потоки могут соответствовать как мобильным объектам, «пропускаемым» через систему (заказы, изделия и их компоненты, перемещаемые грузы), так и ресурсам, поддерживающим жизнедеятельность организационной системы (сырье, энергия, человеческий фактор). Поскольку объектные потоки являются дискретными, для единообразия модели примем, что и ресурсные потоки дискретны, т.е. ресурсы потребляются порциями. Порцию потока будем называть его элементом.

Потоковая модель организационной системы отображает взаимодействие операций с накопителями элементов потоков, заключающееся во внесении операцией элементов в накопитель, изъятии операцией элементов из накопителя, преобразовании элементов внутри накопителя согласно заданному (для данного накопителя) жизненному циклу объектов – элементов потока. Накопители, для которых моделируется только внесение/изъятие элементов потока, будем называть пассивными, а для которых, кроме того, моделируется и реализация жизненного цикла – активными. Последние соответствуют используемой в объектно-ориентированном подходе [1] конструкции «класс», поэтому в контексте будем применять и эту терминологию. Состояние пассивного накопителя в каждый момент времени определяется числом находящихся в нем элементов потока, состояние активного – суммарным числом элементов (мобильных объектов) и фазой жизненного цикла каждого из них.

Условное изображение пассивного и активного накопителя элементов потока дано на рис. 4.1. Заметим, что пассивный накопитель – частный случай активного.

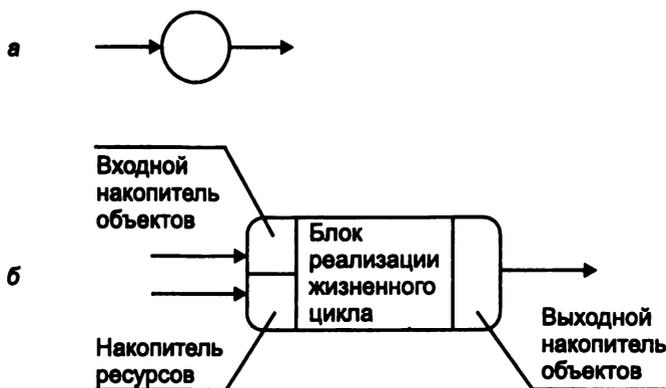


Рис. 4.1. Изображение пассивного (а) и активного (б) накопителя элементов потока

Потоковая модель организационной системы изображается ориентированным графом с вершинами двух типов, сопоставленными с операциями и накопителями, в котором дугами соединяются лишь вершины разного типа (двудольный граф). В настоящем учебном пособии рассматриваются два предельных варианта потоковой модели – на базе только пассивных и только активных накопителей. Эти варианты отличаются примененным математическим аппаратом и определяют два направления, которые будем называть пассивным и активным потоковым моделированием. Пассивное моделирование опирается на аппарат сетей Петри [6, 10], активное – на аппарат эволюционно-потоковых диаграмм [16]. Возможен и промежуточный вариант потоковой модели, основанный на совместном использовании накопителей обоих типов, однако его исследование выходит за пределы данного учебного пособия.

При предпроектном анализе еще не существует реализации системы, она представлена на уровне сценариев функционирования, в том числе потокового сценария [16].

Последний описывается на основе графических средств [16], развивающих нотации технологии структурного анализа и проектирования SADT [9]. В связи с этим возникает задача формального преобразования потокового сценария в реализующую его потоковую модель.

Решение поставленной задачи рассматривается в настоящей главе применительно к многоагентным иерархическим организационным системам (МИОС), где под агентами понимаются автономно функционирующие организационные или технические структуры. В форме МИОС может быть представлена любая сложная экономическая, политическая и иная структура – производственная, административная, торговая, военная и т.д. Например, в производственной корпорации агентом верхнего (нулевого) уровня является руководство корпорации, агентами первого уровня – входящие в ее состав предприятия, агентами второго уровня – цехи, отделы и т.д.

Агенты МИОС реализуют процессы, которые состоят из отдельных целенаправленных операций, выполняемых в определенном порядке. При этом агент  $i$ -го уровня ( $i = 0, 1, \dots, N-1$ , где  $N$  – число уровней) реализует процессы на базе операций, реализуемых агентами  $i + 1$ -го уровня. Иными словами, в МИОС имеет место иерархия процессов, в ходе которых и осуществляется движение потоков.

Отметим характерную особенность МИОС: среди ее уровней существует такой  $k$ -й уровень иерархии, что в процессах вышестоящих уровней ( $i < k$ ) операции выполняются в свободном (произвольном) порядке, зависящем только от складывающейся ситуации, в том числе все операции могут выполняться параллельно (параллельная работа предприятий корпорации, цехов и отделов предприятия и т.д.). В процессах  $k$ -го и всех нижестоящих уровней ( $i \geq k$ ) из-за наличия жестких технологических связей операции выполняются в фиксированном порядке.

В настоящей главе последовательно обсуждается методика:

- формального описания потокового сценария МИОС [16, 18];
- формального перехода от описания сценария к описанию пассивных и активных потоковых моделей МИОС при свободном и фиксированном порядке выполнения операций [19];
- имитации функционирования МИОС на основе пассивной и активной потоковых моделей [16].

Пассивная потоковая модель представляет собой «триадную схему» [18], состоящую из трех взаимодействующих частей – субмоделей (рис. 4.2).

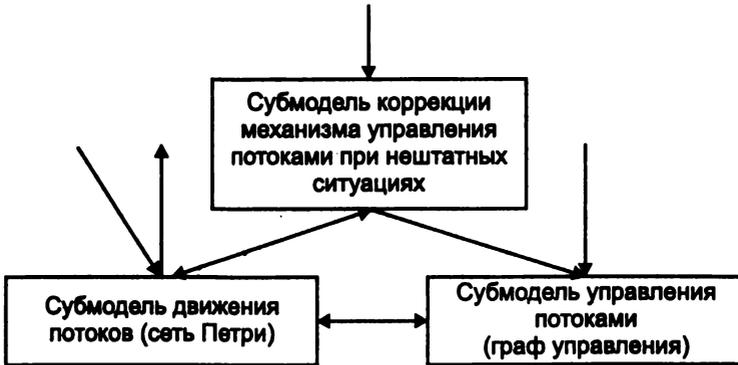


Рис. 4.2. Пассивная потоковая модель (триадная схема)

Для моделирования потоков в системе выделяются «точки наблюдения», соответствующие подпространствам системного пространства, в которых содержится  $n \geq 0$  элементов (порций) потока. В качестве элементов потока могут выступать денежные суммы, комплекты документации, единицы товара, единицы энергии и т.д. Точки наблюдения подразделяются на входные, внутренние и выходные. Субмодель движения потоков описывает все возможные варианты перемещения элементов потоков между точками наблюдения на основе сети Петри, которую будем называть потоковой сетью. Взаимодействие моделируемой системы с внешней средой осуществляется через входные и выходные точки наблюдения.

Субмодель управления потоками, получая на вход информацию из точек наблюдения и информацию о состоянии внешней среды, обеспечивает определенным сценарием порядок выполнения операций, инициирующих изменения в потоках (перемещения элементов потоков между накопителями). Операции при этом могут выполняться как последовательно, так и параллельно, предусматриваются альтернативный выбор запускаемых операций в зависимости от ситуации на входе субмодели управления, синхронизация параллельных операций, отсутствие «конфликтов» между ними и иные свойства потоковых моделей. Субмодель управления потоками выражается графом управления – сетью Петри, дополнительно помеченной описанием условий смены управленческих операций и процедур воздействия на потоки.

Взаимодействия между субмоделями движения потоков и управления потоками (см. рис. 4.2) проявляются в том, что граф управления, постоянно отслеживая состояние потоковой сети Петри и внешней среды, в определенные моменты времени (при наступлении соответствующих событий) формирует воздействия на потоковую сеть, приводящие к изменению ее состояния.

Субмодель коррекции механизма управления фиксирует «нештатные» ситуации в потоковой сети (например, превышение установленного предела брака выпускаемой продукции, недопустимое падение запаса ресурсов и т.д.) и для поиска выхода из сложившегося положения изменяет порядок функционирования графа управления.

Эффективна следующая процедура поиска выхода – управление управлением потоками. Среди управляющих воздействий выделяются так называемые «точки выбора», для которых имеет место альтернативный выбор последующих воздействий. При возникновении «нештатной» ситуации прерывается связь между потоковой сетью и графом управления, последний возвращается в ближайшую предшествующую точку выбора, и в ней принимается иное альтернативное решение о последующих воздействиях. После этого связь между потоковой сетью и графом управления восстанавливается и процесс моделирования продолжается.

Активная потоковая модель представляет собой детализацию потокового сценария с учетом эволюции объектов в ходе выполнения операций и передачи объектов от одних операций к другим. Операциям сценария сопоставляются активные накопители. Эволюция объектов при выполнении операции проявляется в изменении значений их признаков (атрибутов), при переходах между операциями возникают «мутации» – порождение новых признаков и потеря признаков, ставших ненужными. «Мутация» сопровождается также наследованием определенных признаков. Таким образом, понятия эволюции и «мутации» имеют тот же смысл, что и для живых организмов.

Эволюцию объектов при обработке их конкретной операцией будем трактовать как последовательную смену состояний объектов, определяемую жизненным циклом этой операции (активного накопителя). Все объекты, обрабатываемые операцией, имеют одно и то же множество атрибутов, набор значений которых и определяет состояние объекта. Чтобы проследить траекторию (жизненный путь) каждого объекта, в множестве атрибутов

операции выделяется ключевой атрибут (группа атрибутов). Значения ключевых атрибутов при прохождении объектами операционного жизненного цикла не изменяются.

Помимо эволюционных преобразований объектов операция осуществляет накопление и хранение поступающих на нее (входных) и обработанных ею (выходных) объектов, а также потребление необходимых для ее жизнедеятельности ресурсов. В целом операция представляет собой комплекс, в котором взаимосвязаны (инкапсулированы) информационные данные и процедуры их обработки. Поскольку согласно парадигме объектно-ориентированного подхода операция (активный накопитель) соответствует понятию «класс», то экземплярами класса являются «живущие» в нем объекты.

В упомянутые выше взаимодействия между классами, сопровождаемые мутациями признаков-атрибутов, может быть вовлечено несколько входных и выходных классов. При этом из каждого входного класса может наследоваться (во все выходные классы) «свое» подмножество атрибутов, а в каждом выходном классе порождаться «свое» подмножество атрибутов.

Изложенное выше подводит нас к тому, чтобы в качестве формальной основы для описания активной потоковой модели применить сеть Петри, позиции которой сопоставлены классам, а переходы – межклассовым взаимодействиям. Такое формальное представление активной потоковой модели будем называть эволюционно-потоковой диаграммой [16].

На основе потоковой модели проводятся аналитические и имитационные исследования. При аналитическом исследовании методами теории сетей Петри выявляются и устраняются аномалии при работе системы: «тупики» («зависания»); циклы, не имеющие выхода («ловушки»); повторные вызовы операций во время их выполнения и т.д. При имитационном исследовании формируются протоколы, отображающие динамику потоков в системе (описание протоколов имитационного потокового моделирования дается в виде так называемых конвейерно-временных диаграмм [16]). Аналитическое и имитационное моделирование дополняют друг друга. Первое дает картину аномалий, однако требует из-за своей трудоемкости существенных вычислительных затрат. Второе позволяет получить адекватное представление о движении потоков в системе, однако не гарантирует рассмотрения всех возможных вариантов поведения.

## 4.2.

### Описание потокового сценария организационных систем

Потоковый сценарий МИОС формируется в виде иерархии потоковых сценариев процессов, состоящих из операций.

Графическое описание процесса состоит из наружной рамки (в виде прямоугольника со скругленными углами), внутри которой размещаются изображения операций и потоков, связывающих операции между собой и процесс с внешней средой. Потоки изображаются стрелками, помеченными соответствующим текстом. Содержимым потоков могут быть как ресурсы, поддерживающие жизнедеятельность процесса, так и мобильные объекты, поступающие на вход процесса, «проходящие» через процесс и снимаемые с его выхода.

Слева к рамке процесса подводятся наружные входные стрелки  $P_i$ , помеченные именами входных объектов, и стрелки  $R_j$ , помеченные именами ресурсов. Справа из рамки выводятся наружные стрелки  $P_k$  для выходных объектов.

Стрелки (потоки) внутри рамки, называемые внутренними, подразделяются на входные (свободное начало, конец соединен с изображением операции), выходные (начало соединено с операцией, конец свободный) и промежуточные (соединяют две операции). Каждой входной наружной стрелке сопоставляется  $n \geq 1$  одинаково помеченных с ней входных внутренних стрелок. Это означает, что поданный на вход процесса объектный (ресурсный) поток передается на входы  $n$  операций. Если при этом поток всегда копируется в  $n$  экземплярах, то соответствующие входные внутренние стрелки помечаются только обозначением потока. Если передача потока на вход какой-нибудь из  $n$  операций обусловлена ситуацией, то соответствующее условие вводится в пометку внутренней стрелки (например,  $P_i$ , если  $F_h = 1$ , где  $F_h$  – логическая функция).

Каждой выходной наружной стрелке сопоставляется только одна выходная внутренняя стрелка, т.е. на выход процесса поступает поток от одной операции (иначе процесс противоречив). Деление потока на части, а также сборка его из частей реализуются специальными операциями.

Графическое описание процесса проиллюстрируем на примере конвейерной технологической линии, схема которой приведена

на рис. 4.3. Линия обрабатывает элементы потока – мобильные объекты. Для простоты ресурсы в описании не учитываются.



Рис. 4.3. Схема конвейерной технологической линии

В качестве объектов могут выступать материальные изделия, документы, информация в электронном виде и т.д. Система представляет собой «поточную линию» из трех последовательных позиций  $q_1$ ,  $q_2$ ,  $q_3$ , на которых проводятся соответственно обработка 1, обработка 2, контроль и сортировка объектов на годные (сертифицированные) и отбракованные. Годные объекты передаются на выходную позицию  $q_4$ , отбракованные – на позицию  $q_5$ . После завершения обработки объектов на позициях  $q_1$ ,  $q_2$ ,  $q_3$  они синхронно передаются на следующую позицию: из  $q_1$  на  $q_2$ , из  $q_2$  на  $q_3$ , из  $q_3$  в зависимости от результатов контроля на  $q_4$  или  $q_5$ . Будем различать три типа объектов: объекты типа 1 подвергаются только обработке 1, типа 2 – только обработке 2, типа 3 – последовательно обработке 1 и обработке 2. Графическое описание потокового сценария линии дано на рис. 4.4.

Операции (процессы), реализуемые на позициях  $q_1$ ,  $q_2$ ,  $q_3$ , обозначены соответственно через  $\Pi_1$ ,  $\Pi_2$ ,  $\Pi_3$ . Рядом с изображением операций помещаем обозначение агента, выполняющего операцию (в нашем случае – позицию  $q_i$ ), которое заключаем в кружок. Аналогично нотации SADT [9] над изображением операции могут указываться ее цели (на рис. 4.4 для простоты опущены).

Объектам типа 1, 2, 3 в сценарии соответствуют входные потоки  $P_1$ ,  $P_2$ ,  $P_3$ . Потоки  $P_1$ ,  $P_3$  передаются на вход операции  $\Pi_1$ , поток  $P_2$ , – на вход  $\Pi_2$ . Операция  $\Pi_1$  преобразует потоки  $P_1$ ,  $P_3$  соответственно в промежуточные потоки  $P_4$ ,  $P_5$ , а операция  $\Pi_2$  – входной поток  $P_2$  и промежуточный  $P_5$  – в промежуточные потоки соответственно  $P_6$ ,  $P_7$ . Операция  $\Pi_3$  получает на вход потоки  $P_4$ ,

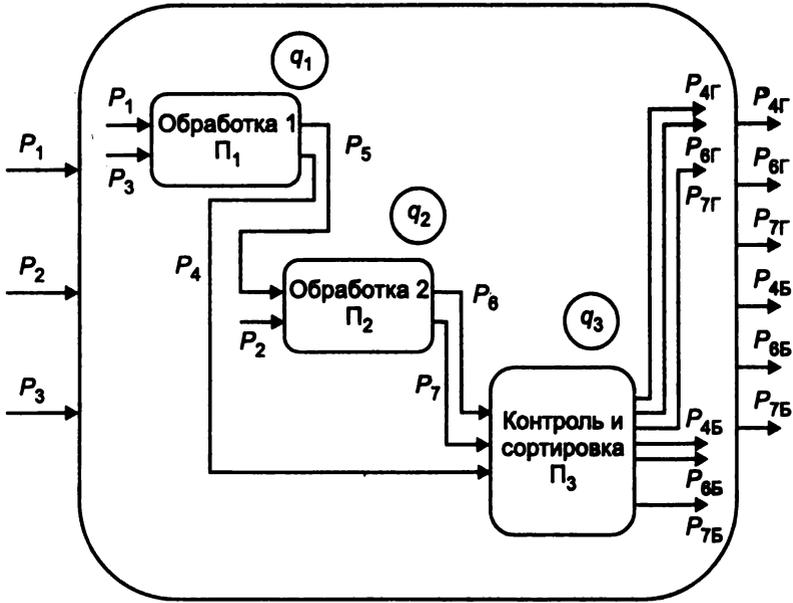


Рис. 4.4. Описание потокового сценария конвейерной технологической линии

$P_6$ ,  $P_7$  и формирует выходные потоки годных объектов  $P_{4Г}$ ,  $P_{6Г}$ ,  $P_{7Г}$  и бракованных объектов  $P_{4Б}$ ,  $P_{6Б}$ ,  $P_{7Б}$ .

Преобразование потоков при реализации операций выражается логическими формулами, в которых стрелки  $i_j$  представляют отображение набора входных потоков операции в набор ее выходных потоков. Если формула содержит несколько стрелок, то соответствующие им подформулы (заклучены в скобки) соединяются логической связкой ИЛИ.

Логика преобразования потоков для сценария, описанного на рис. 4.4, такова:

Операция	Логическая формула
$\Pi_1$	$(P_1 \xrightarrow{i_1} P_4) \vee (P_3 \xrightarrow{i_2} P_5)$
$\Pi_2$	$(P_2 \xrightarrow{i_3} P_6) \vee (P_5 \xrightarrow{i_4} P_7)$
$\Pi_3$	$(P_4 \xrightarrow{i_5} P_{4Г}) \vee (P_4 \xrightarrow{i_6} P_{4Б}) \vee (P_6 \xrightarrow{i_7} P_{6Г}) \vee (P_6 \xrightarrow{i_8} P_{6Б}) \vee (P_7 \xrightarrow{i_9} P_{7Г}) \vee (P_7 \xrightarrow{i_{10}} P_{7Б})$

В качестве другого примера потокового сценария МИОС рассмотрим *компанию по обслуживанию заказов*.

Компания по обслуживанию заказов принимает от клиентов заявки на заказы, комплектует их из запасов товаров на складе компании, при необходимости пополняет эти запасы путем обращения к производителям, доставляет сформированные заказы клиентам. Фрагмент организационной структуры компании показан на рис. 4.5, где агент  $A_0$  представляет руководство компании; агенты  $A_1, A_2, A_3$  – соответственно отделы приема заявок на заказы, комплектации заказов, доставки заказов; агенты  $A_{1.1}, A_{1.2}$  – соответственно подотделы контроля заявок и финансовых расчетов.

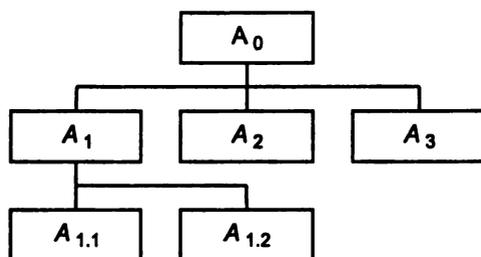


Рис. 4.5. Фрагмент организационной структуры компании по обслуживанию заказов

Фрагмент графического описания потокового сценария функционирования компании дан на рис. 4.6. Он представляет собой трехуровневую иерархию процессов  $\Pi_0, \Pi_1, \Pi_{1.1}$ , где процесс  $\Pi_0$  реализуется на базе операций, выполняемых агентами  $A_1, A_2, A_3$  (см. рис. 4.5), процесс  $\Pi_1$  детализирует соответствующую операцию в  $\Pi_0$  и выполняется агентами  $A_{1.1}, A_{1.2}$ , процесс  $\Pi_{1.1}$ , детализирующий операцию в  $\Pi_1$ , состоит из операций агента  $A_{1.1}$ , которые в сценарии не детализируются (терминальные операции). Порядок выполнения операций в процессах  $\Pi_0, \Pi_1$  является свободным, в процессе  $\Pi_{1.1}$  – фиксированным.

Проиллюстрируем потоковый сценарий на примере процесса  $\Pi_0$  (см. рис. 4.6). На вход процесса подаются поток заявок  $P_1$  и информационный поток  $R_1$ , отображающий состояние ресурсов – запас товаров на складе компании. Поток  $P_1$  поступает на вход операции  $\Pi_1$  – *прием заявок и оплата заказов*, реализуемой аген-

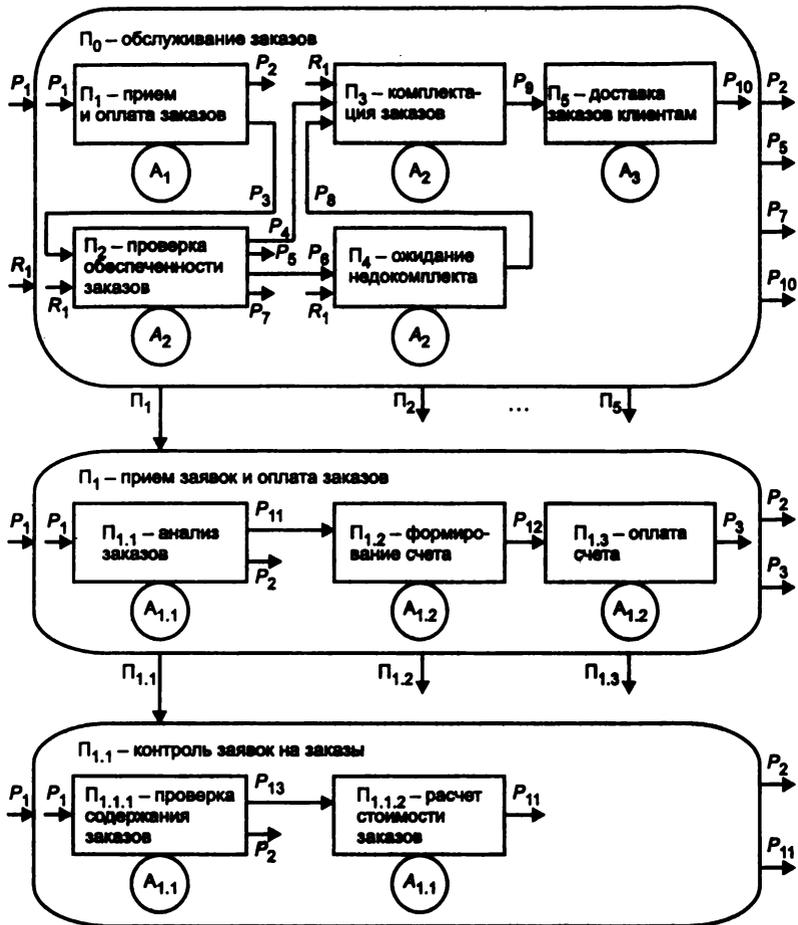


Рис. 4.6. Фрагмент потокового сценария для МИОС, показанной на рис.4.5

том  $A_1$ . Поток  $R_1$  поступает на вход операций  $\Pi_2$  – проверка обеспеченности заказов,  $\Pi_3$  – комплектация заказов,  $\Pi_4$  – ожидание поступления недокомплекта, реализуемых агентом  $A_2$ . Операция  $\Pi_1$  либо отклоняет заявку (поток  $P_2$ , передаваемый на выход процесса  $\Pi_0$ ), либо принимает заявку, осуществляет оплату и направляет данные заказа операции  $\Pi_2$  (промежуточный поток  $P_3$ ). Опе-

рация  $\Pi_2$  определяет, обеспечен ли заказ ресурсами  $R_1$ . Если да, то на склад направляется требование на бронирование соответствующих товаров (выходной поток  $P_5$ ), а данные об обеспеченном заказе пересылаются операции  $\Pi_3$  (промежуточный поток  $P_4$ ). Если нет, то направляется заявка производителю товара (выходной поток  $P_7$ ), а данные необеспеченного заказа направляются операции  $\Pi_4$  (промежуточный поток  $P_6$ ).

Операция  $\Pi_4$  непрерывно проверяет, достаточен ли ресурс  $R_1$  для комплектования ожидающих заказов. Если некоторые из них становятся обеспеченными (подготовленными), то они передаются операции  $\Pi_3$  (промежуточный поток  $P_8$ ), так же как и заказы из потока  $P_4$ . Операция  $\Pi_3$  изымает из  $R_1$  необходимые товары и направляет укомплектованные заказы на операцию  $\Pi_5$  – доставка заказов клиентам, реализуемую агентом  $A_3$  (промежуточный поток  $P_9$ ). Операция  $\Pi_5$  формирует выходной поток  $P_{10}$  реализованных (доставленных) заказов.

Приведем логические формулы, отображающие преобразование потоков операциями процесса  $\Pi_0$  (см. рис. 4.6).

Операция	Логическая формула
$\Pi_1$	$(R_1 \xrightarrow{t_1} P_2) \vee (R_1 \xrightarrow{t_2} P_3)$
$\Pi_2$	$(P_3 R_1 \xrightarrow{t_3} P_4 P_3 R_1) \vee (P_3 R_1 \xrightarrow{t_4} P_6 P_7 R_1)$
$\Pi_3$	$(P_4 R_1 \xrightarrow{t_5} P_9) \vee (P_8 R_1 \xrightarrow{t_6} P_9)$
$\Pi_4$	$P_6 R_1 \xrightarrow{t_7} P_8 R_1$
$\Pi_5$	$P_9 \xrightarrow{t_8} P_{10}$

Заметим, что в формулах, соответствующих операциям  $\Pi_2$  и  $\Pi_4$ , поток  $R_1$  является и входным, и выходным. Это отражает тот факт, что при выполнении операций  $\Pi_2$ ,  $\Pi_4$  запас  $R_1$  товаров на складе учитывается, но не расходуется.

Смежные процессы (принадлежащие  $i$ -му и  $i+1$ -му уровням потокового сценария) должны быть согласованы по входным и выходным потокам. Это проявляется в том, что внутренние вход-выходные потоки операции в процессе  $i$ -го уровня наследуются один к одному соответствующим процессом  $i+1$ -го уровня (на рис. 4.6 процесс  $\Pi_1$  наследует внешний вход  $P_1$  и внешние выходы  $P_2$ ,  $P_3$  от операции  $\Pi_1$  процесса  $\Pi_0$ , аналогичное имеет место для про-

цесса  $\Pi_{i+1}$  и операции  $\Pi_{i+1}$  в процессе  $\Pi_i$ ). В общем случае наряду с наследуемыми процесс  $i+1$ -го уровня может иметь и дополнительные (свои) внешние потоки, которые наследуются нижестоящими уровнями (начиная с  $i+2$ -го уровня).

Как уже отмечалось во введении, предлагаемый здесь язык описания потоковых сценариев МИОС является развитием широко известной технологии SADT [9], что заключается в:

- применении более простого и наглядного способа согласования смежных уровней сценария;
- дополнении графического описания процессов аналитическим описанием логики преобразования потоков при реализации операций.

В SADT-технологии устанавливается соответствие (как правило, не взаимно однозначное) между стрелками, принадлежащими разным графическим схемам – описаниям смежных процессов. В предлагаемом языке это осуществляется внутри описания процесса – в виде соответствия между наружными и внутренними входными (выходными) стрелками. Разработчику сценария легче оперировать с одной графической схемой, чем одновременно с несколькими.

### 4.3.

#### **Пассивное потоковое моделирование организационных систем**

Потоковый сценарий МИОС, представленный графическим описанием (см. рис. 4.4 и 4.6) и логическими формулами преобразования потоков (см. с. 98, 101), моделируется путем последовательного анализа и/или имитации функционирования триадных схем, сопоставленных процессам. Каждая схема моделируется локально (независимо от других схем), после чего проверяется их согласованность. Навигация по сценарию может осуществляться как сверху вниз, так и снизу вверх. Локальное моделирование (по частям) является эффективным средством преодоления сложности, так как последовательно работать с несколькими схемами ограниченной размерности (до 10 компонент) проще, чем с одной сложной схемой.

Процедура пассивного потокового моделирования процесса, реализуемого в МИОС, состоит из следующих этапов.

1. *Построение субмодели движения потоков (потокосетной сети)* – сети Петри, позиции которой сопоставлены внутренним стрелкам  $P_i$  в сценарии процесса, а переходы – стрелкам  $t_i$  в логических формулах преобразования потоков. При этом выполняются шаги [16]:

1.1. Для каждой операции процесса удаляется каждая входная (выходная) стрелка  $P_i$  и вместо нее внутри рамки операции помещается входная (выходная) позиция, помеченная так же, как и удаленная стрелка.

1.2. Каждой стрелке  $t_i$  в логической формуле преобразования потоков данной операции сопоставляется переход, также размещаемый внутри рамки операции.

1.3. Введенный переход  $t_i$  соединяется с входными и выходными позициями операции согласно логической формуле.

1.4. Одинаково помеченные позиции, принадлежащие разным операциям, отождествляются.

2. *Построение субмодели управления потоками (графа управления)* – сети Петри, позиции которой соответствуют управленческим действиям  $u_i$ , а переходы – событиям  $S_i$ , регистрирующим изменения в потокосетной сети и инициирующим следующие управленческие воздействия.

Выполнение управленческих действий отображается наличием информационного маркера (изображаемого жирной точкой) в соответствующей позиции графа, причем в позиции может находиться не более одного маркера. Завершение управленческих действий при моделировании процесса определяется истечением времени, отсчитываемого виртуальным операционным таймером. Длительность действия либо задается извне, либо вычисляется на основе ситуации, сложившейся в потокосетной сети (маркирование ее позиций).

Для процессов со свободным порядком выполнения операций для каждой операции вводится свой граф управления. Это дает потенциальную возможность параллельного выполнения всех операций сценария. При фиксированном порядке выполнения операций (нижние уровни МИОС) используется единый граф управления всеми операциями, отображающий этот порядок.

3. *Организация взаимодействия между потокосетной сетью и графом управления* (рис. 4.7). Переходы потокосетной сети и графа управления дополнительно помечаются внешними логическими условиями их срабатывания. Для потокосетной сети это обозначения управленческих воздействий, для графов управления – логические

функции, отображающие состояние потоковой сети (число маркеров в ее позициях). Воздействия, описываемые указанными условиями, подаются на вход субмоделей (горизонтальные стрелки между потоковой сетью и графами управления на рис. 4.7). При этом реализуется следующее двустороннее взаимодействие: воздействием со стороны графа управления запускаются определенные переходы потоковой сети, инициируя изменение ее состояния. Результатом изменения является запуск следующих переходов на графе управления и т.д.

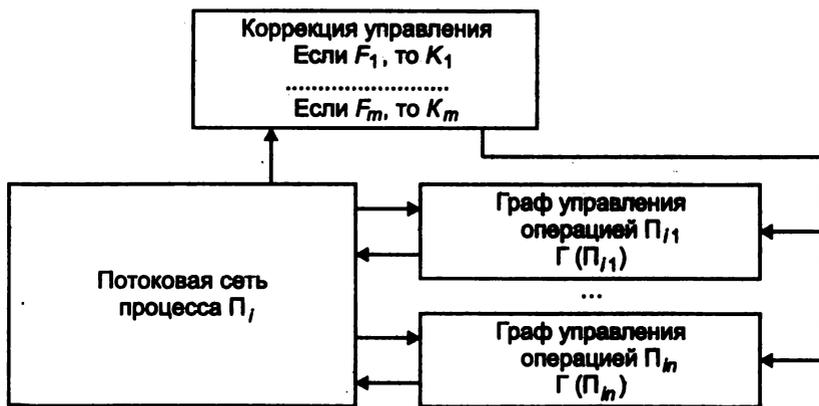


Рис. 4.7. Триадная потоковая схема для процесса со свободным порядком выполнения операций

4. *Коррекция управления в триадной потоковой схеме* (см. рис. 4.7) описывается набором условных выражений: если  $F_i$ , то  $K_i$ ,  $i = 1, \dots, m$ , где  $F_i$  – логическое условие возникновения «нештатной» ситуации в потоковой сети,  $K_i$  – процедура изменения условий срабатывания переходов на графах управления операциями процесса. Блок коррекции «наблюдает» за взаимодействием потоковой сети и графов управления и «вмешивается» при «нештатных» ситуациях.

Триадная потоковая схема для процесса с фиксированным порядком выполнения операций, как уже отмечалось, отличается от приведенной на рис. 4.7 тем, что в ней используется единый граф управления всеми операциями (отображающий указанный

фиксированный порядок). Схему с фиксированным порядком можно рассматривать как частный случай базовой схемы (см. рис. 4.7).

5. *Проведение моделирования триадной потоковой схемы на основе аналитических и имитационных методов.* Аналитические методы, в том числе изложенные в гл. 3 как аппарат операционного моделирования, применяют к компонентам триадной схемы – потоковой сети и графам управления. Функционирование триадной схемы в целом, определяемое взаимодействием всех трех ее компонентов, моделируется имитационными методами. Результатом имитационного моделирования являются конвейерно-временные диаграммы (КВД), отображающие изменение состояния мобильных объектов, «пропускаемых» через процесс, и состояния ресурсов процесса на временной шкале моделирования.

Процедуру пассивного потокового моделирования проиллюстрируем на двух примерах.

*Пример 1. Конвейерная технологическая линия.* Потоковая сеть, построенная согласно шагу 1 на основе сценария (см. рис. 4.4) и логических формул (см. с. 98), состоит из трех не связанных между собой фрагментов, представленных на рис. 4.8.

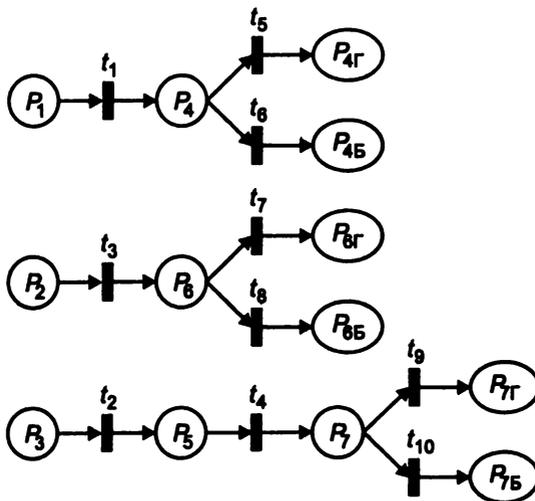


Рис. 4.8. Субмодель движения потоков в конвейерной технологической линии

Операции конвейерной линии, определяющие функционирование потоковой сети, выполняются в фиксированном порядке. Поэтому строится единый граф управления. Граф управления потоками, полученный на шаге 2, показан на рис. 4.9, содержимое управленческих действий дано в табл. 4.1, описание условий срабатывания переходов приведено ниже.

Обозначение перехода	Условие срабатывания перехода
$S_1$	Таймер $V_n$ запущен
$S_2$	Истечение времени $V(u_5)$
$S_3$	[Истечение времени $\max(V(u_2), V(u_3), V(u_4)) \wedge [\max(V(u_2), V(u_3), V(u_4)) > V_n]$
$S_4$	[Истечение времени $\max(V(u_2), V(u_3), V(u_4)) \wedge [\max(V(u_2), V(u_3), V(u_4)) \leq V_n]$
$S_5$	[Индекс $G$ вычислен] $\wedge [G \leq G_n]$
$S_6$	[Индекс $G$ вычислен] $\wedge [G > G_n]$
$S_7$	[Проведена коррекция $V_n$ ] $\wedge$ [таймер $V_n$ сброшен]

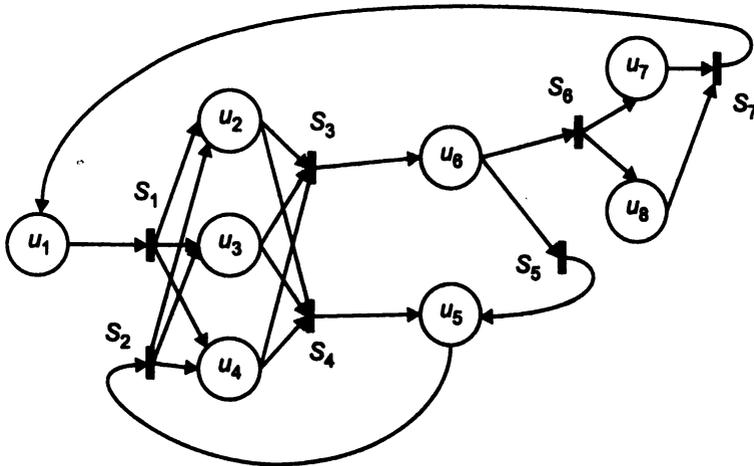


Рис. 4.9. Граф управления конвейерной технологической линией

Рассмотрим работу графа управления и его взаимодействие с потоковой сетью.

Срабатывание в графе управления перехода  $S_1$ , активизированного в начальной маркировке, происходит по факту запуска таймера  $V_m$ . Маркер удаляется из позиции  $u_1$ , а в позиции  $u_2, u_3, u_4$  графа управления вносится по маркеру – начинается параллельное выполнение операций в позициях  $q_1, q_2, q_3$  конвейерной линии (см. рис. 4.3) и поступает разрешение на срабатывание переходов потоковой сети (см. рис. 4.8).

Управленческое действие  $u_2$  разрешает переход  $t_1$ , если число маркеров в позиции  $P_1$  потоковой сети (количество объектов типа 1 в накопителе на входе линии) не меньше числа маркеров в позиции  $P_3$  (количества объектов типа 3), т.е.  $P_1 \geq P_3$ , либо разрешает переход  $t_2$  в противном случае (при  $P_1 < P_3$ ). Аналогичное применительно к позициям  $P_2, P_5$  и переходам  $t_3, t_4$  имеет место при действии  $u_3$ .

Таблица 4.1

Содержание управленческих действий

Обозначение действия	Содержание действия	Разрешаемые переходы потоковой сети
$u_1$	Запуск отсчета времени моделирования $V_m$	–
$u_2$	Обработка 1 на позиции $q_1$	Если $P_1 \geq P_3$ , то $t_1$ , если $P_1 < P_3$ , то $t_2$
$u_3$	Обработка 2 на позиции $q_2$	Если $P_2 \geq P_5$ , то $t_3$ , если $P_2 < P_5$ , то $t_4$
$u_4$	Контроль и сортировка на позиции $q_3$	Если $\max(P_4, P_6, P_7) = P_4$ , то $t_5 \vee t_6$ Если $\max(P_4, P_6, P_7) = P_6$ , то $t_7 \vee t_8$ Если $\max(P_4, P_6, P_7) = P_7$ , то $t_9 \vee t_{10}$
$u_5$	Передача объектов между позициями $q_1, \dots, q_5$	–
$u_6$	Вычисление индекса $G$ допустимых прерываний	–
$u_7$	Корректировка критического времени $V_k$	–
$u_8$	Сброс времени моделирования $V_m$	–

При выполнении управленческого действия  $u_4$  разрешен один из переходов  $t_5$  или  $t_6$ , если число маркеров в позиции  $P_4$  превышает число маркеров в  $P_6, P_7$  (разрешен  $t_7$  или  $t_8$ , если наибольшее число маркеров содержится в  $P_6; t_9$  или  $t_{10}$ , если в  $P_7$ ). Выбор одного из двух переходов, а также выбор позиции при совпадении  $\max(P_4, P_6, P_7)$  имеет вероятностный характер и определяется генератором случайных чисел. Действия  $u_5, \dots, u_8$  (см. табл. 4.1) не вызывают изменений в потоковой сети.

Описанные условия разрешения срабатывания переходов потоковой сети (правый столбец табл. 4.1) отражают одну из возможных дисциплин и даны лишь в качестве примера. Возможны и другие дисциплины воздействия графа управления на потоковую сеть.

По истечении времени  $\max(V(u_2), V(u_3), V(u_4))$ , наиболее продолжительного из параллельных действий  $u_2, u_3$  и  $u_4$ , граф управления принимает решение:

Если  $\max(V(u_2), V(u_3), V(u_4)) \leq V_k$ , то срабатывает переход  $S_4$ ,

Если  $\max(V(u_2), V(u_3), V(u_4)) > V_k$ , то срабатывает переход  $S_3$ ,

где  $V_k$  – критическое время рабочего цикла конвейерной линии;

$V(u_i)$  – длительность действия  $u_i$ .

Выбор при моделировании первой или второй альтернативы является случайным событием.

При срабатывании  $S_4$  удаляются маркеры из позиций  $u_2, u_3, u_4$  графа управления, вносится маркер в его позицию  $u_5$  и инициируется операция передачи объектов между позициями  $q_1, \dots, q_5$  конвейерной линии (см. рис. 4.3). Окончание действия  $u_5$  контролируется по времени  $V(u_5)$  с помощью операционного таймера, по истечении  $V(u_5)$  в графе управления срабатывает переход  $S_2$ , маркер удаляется из  $u_5$  и вносятся маркеры в  $u_2, u_3, u_4$ . Рабочий цикл графа управления повторяется.

Если продолжительность рабочего такта превысила критическое время и сработал переход  $S_3$  графа управления, то рабочий цикл прерывается и запускается действие  $u_6$  вычисления индекса допустимых прерываний  $G = N/V_m$ , где  $N$  – суммарное число прерываний за время моделирования  $V_m$ . Результатом действия  $u_6$  может быть либо срабатывание перехода  $S_5$ , если  $G \leq G_n$ , где  $G_n$  – нормативное допустимое значение индекса, либо срабатывание  $S_6$ , если  $G > G_n$ . В первом случае возобновляется рабочий цикл с дей-

ствия  $u_5$ , во втором – запускаются параллельные действия  $u_7$  (коррекция критического времени  $V_k$  рабочего цикла) и  $u_8$  (сброс таймера отсчета времени моделирования  $V_m$ ).

Действие  $u_7$  реализуется в субмодели коррекции механизма управления потоками при нештатных ситуациях (см. рис. 4.2) и будет описано ниже.

По завершении действий  $u_7$  и  $u_8$  при выполнении условия [проведена коррекция  $V_k$ ] / [таймер  $V_m$  сброшен] срабатывает переход  $S_7$  и функционирование графа управления возобновляется с действия  $u_1$ .

О качестве работы конвейерной линии будем судить по показателю  $G$  – индексу допустимых прерываний, который, в свою очередь, зависит от критического времени  $V_k$  рабочего цикла. Зависимость  $G = f(V_k)$  выражается следующим образом: если  $V_k$  возрастает, то более вероятно выполнение условия  $V \leq V_k$ , запускающего переход  $S_4$  графа управления, чем условия  $V > V_k$ , запускающего переход  $S_3$ . Это означает, что при увеличении  $V_k$  уменьшается вероятность прерывания, т.е. улучшается качество работы линии за счет сокращения простоев (при уменьшении  $V_k$  вероятность прерывания увеличивается).

Пусть при моделировании линии критическое время имело начальное значение  $V_{k.нач}$ . Сформулируем оптимизационную задачу: найти минимальное значение критического времени  $V_{k.мин}$ , при котором выполняется условие  $G \leq G_n$ . Эта задача решается действием  $u_7$  (см. рис. 4.9) на основе следующего алгоритма.

1. При первом прерывании проверить условие  $G \leq G_n$ .
2. Если условие выполняется, то проверяем, не завышено ли значение  $V_{k.нач}$ , т.е. можно ли его уменьшить. Для этого полагаем  $V_k = V_{k.нач} - 1$  и продолжаем моделирование. Если за установленное время не произойдет прерывания, то принимаем  $V_{k.мин} = V_{k.нач} - 1$ . В противном случае повторяем шаг 2.
3. Если на шаге 1  $G > G_n$ , то выполняем присваивание  $V_k = V_{k.нач} + 1$  и возобновляем моделирование. При выполнении условия  $G \leq G_n$  и отсутствии прерываний (за установленное время) принимаем  $V_{k.мин} = V_{k.нач} + 1$ , при возникновении прерывания повторяем шаг 3 до тех пор, пока не будет найдено минимальное критическое время, удовлетворяющее условию задачи.

При описании результатов имитационного моделирования конвейерной технологической линии объекты, пропускаемые че-

рез линию, будем обозначать через  $x_{ij}$ , где  $i = 1, \dots, N$  – номер объекта,  $j = 1, 2, 3$  – его тип. Объекты реализуют различные маршруты движения между позициями линии  $q_k$ ,  $k = 1, \dots, 5$  (см. рис. 4.3). Множество всех возможных маршрутов дано ниже.

Тип объектов	Маршрут
1	$q_1 q_3 q_4$
	$q_1 q_3 q_5$
2	$q_2 q_3 q_4$
	$q_2 q_3 q_5$
3	$q_1 q_2 q_3 q_4$
	$q_1 q_2 q_3 q_5$

Момент прихода объекта с номером  $i$  на позицию  $q_k$  обозначим  $\Theta_i(q_k)$ .

Конвейерно-временная диаграмма, полученная при имитационном моделировании линии, представлена на рис. 4.10. На ней изображена временная шкала моделирования  $\tau$  (ось времени) и под ней шкалы для объектов  $x_{ij}$ , отображающие временной интервал пребывания объектов в моделируемой системе. Моменты  $\Theta_i(q_k)$ , показанные на объектных шкалах кружками, проектируются на ось времени в виде последовательности моментов  $\tau_1, \dots, \tau_m$ . Полагаем, что объекты  $x_{ij}$  поступают на линию в очередности, определяемой их номером  $i$ , поэтому соответствующие шкалы сдвинуты одна относительно другой вправо. Из диаграммы видно, какие объекты в каждый момент времени обрабатываются на линии параллельно.

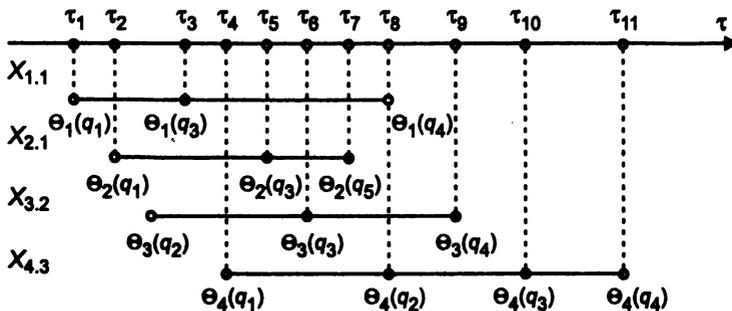


Рис. 4.10. Конвейерно-временная диаграмма прохождения объектов через технологическую линию

**Пример 2. Компания по обслуживанию заказов.** Моделирование проведем на примере процесса  $\Pi_0$ , образующего верхний уровень потокового сценария МИОС (см. рис. 4.6). Поточковая сеть для процесса  $\Pi_0$  с учетом логических формул преобразования потоков (см. разд. 4.2) дана на рис. 4.11. Группы переходов сети, сопоставленные операциям сценария  $\Pi_1, \dots, \Pi_5$ , на рисунке выделены пунктиром.

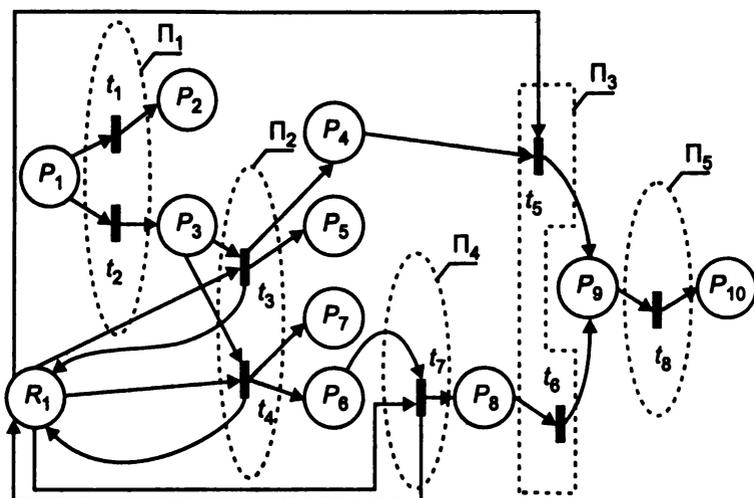


Рис. 4.11. Поточковая сеть процесса  $\Pi_0$

Операции  $\Pi_i, i = 1, \dots, 5$ , выполняются в процессе  $\Pi_0$  в свободном порядке, поэтому для каждой операции  $\Pi_i$  вводим согласно схеме (см. рис. 4.7) свой граф управления  $\Gamma(\Pi_i)$ . Граф  $\Gamma(\Pi_1)$  управления операцией  $\Pi_1$  показан на рис. 4.12.

Проиллюстрируем взаимодействие между потоковой сетью (см. рис. 4.11) и графом  $\Gamma(\Pi_1)$  управления операцией  $\Pi_1$  (рис. 4.12). Содержание действий, соотнесенных позициям графа  $\Gamma(\Pi_1)$ , приведено в табл. 4.2. Условия срабатывания его переходов следующие:

Обозначение перехода	Условие срабатывания перехода
$S_1$	<Позиция графа потоков не пуста>
$S_2$	<Истечение времени действия $u_1$ > $\wedge$ <Выделенный заказ отклонен>

Обозначение  
перехода

Условие срабатывания перехода

$S_3$	$\langle \text{Истечение времени действия } u_1 \rangle \wedge \langle \text{Выделенный заказ принят и оплачен} \rangle$
$S_4$	$P_2(\tau) = P_2(\tau-1) + 1$
$S_5$	$P_3(\tau) = P_3(\tau-1) + 1$

При нахождении маркера в позиции  $u_2$  графа  $\Gamma(\Pi_1)$  запускается переход  $t_1$  в потоковой сети, при нахождении в  $u_3$  запускается переход  $t_2$ . Результатом срабатывания в текущий момент перехода  $t_1$  (перехода  $t_2$ ) является изменение состояния потоковой сети: число маркеров в позиции  $P_2$  (позиции  $P_3$ ) увеличилось на единицу по сравнению с предыдущим моментом  $\tau-1$ , т.е.  $P_2(\tau) = P_2(\tau-1) + 1$  ( $P_3(\tau) = P_3(\tau-1) + 1$ ). Это изменение сообщается графу  $\Gamma(\Pi_1)$  и инициирует в нем переход  $S_4$  (переход  $S_5$ ), переводящий маркер в исходную позицию  $u_0$ .

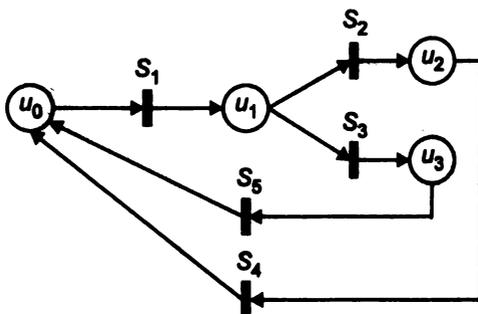


Рис. 4.12. Граф  $\Gamma(\Pi_1)$  управления операцией  $\Pi_1$

Таблица 4.2

Управленческие действия графа  $\Gamma(\Pi_1)$

Обозначение действия	Содержание управленческого действия	Разрешаемые переходы графа потоков
$u_0$	Процесс $\Pi_1$ не активирован	—
$u_1$	Выделить заказ $P_{1i}$ с максимальным временем нахождения в позиции $P_1$ и определить состояние $P_{1i}$	—

Обозначение действия	Содержание управленческого действия	Разрешаемые переходы графа потоков
$u_2$	Передать сообщение «заказ $P_{1i}$ отклонен»	$t_1$
$u_3$	Передать сообщение «заказ принят и оплачен»	$t_2$

Коррекция управления потоками при «нештатных» ситуациях, а также построение конвейерно-временной диаграммы были обсуждены в примере 1, а в примере 2 опущены.

Подведем итоги изложенной технологии пассивного потокового моделирования МИОС.

Основные положения технологии заключаются в следующем.

1. Функционирование МИОС описывается иерархией взаимосогласованных процессов, реализуемых агентами системы – автономными организационными либо техническими структурами. Процесс состоит из целенаправленных действий – операций, выполняемых в определенном порядке агентами нижестоящего уровня.

В процессах высоких уровней иерархии операции выполняются в свободном (произвольном) порядке, зависящем только от складывающейся ситуации, в том числе все операции могут выполняться параллельно. В процессах низших уровней иерархии из-за наличия жестких технологических связей операции выполняются в фиксированном порядке.

2. Сценарий динамики потоков – потоковый сценарий МИОС моделируется путем последовательного имитационного и/или аналитического исследования функционирования процессов. При этом каждый процесс моделируется локально (независимо от других процессов), после чего проверяется их согласованность. Локальное моделирование (по частям) является эффективным средством преодоления сложности, так как последовательно работать с несколькими системами ограниченной размерности (до 10 компонент) проще, чем с одной сложной системой.

3. Описание потокового сценария МИОС проводится на основе изобразительных средств, развивающих технологию структурного анализа и проектирования SADT [9]. Развитие SADT заключается:

- во введении графических нотаций, обеспечивающих более простой и наглядный способ согласования смежных уровней сценария;

- в дополнении графического описания процессов логическими формулами преобразования потоков при реализации операций.

4. Сценарное описание процессов преобразуется в триадные потоковые схемы, представляющие собой структурную модель реализации процессов. Триадная схема состоит из трех взаимодействующих блоков, отображающих соответственно динамику потоков, управление потоками и коррекцию управления при возникновении нештатных ситуаций. Она является функционально распределенной моделью, «прозрачной» и удобной при предпроектном анализе.

Предложена технология формирования на основе сценария триадных потоковых схем процессов при свободном и фиксированном порядке выполнения операций.

#### **4.4.**

### **Активное потоковое моделирование организационных систем**

Активное моделирование начинают с преобразования потокового сценария в эволюционно-потоковую диаграмму (ЭПД). Последняя, как уже отмечалось в разд. 4.1, представляет собой сеть Петри, позиции которой, называемые классами, соответствуют операциям процесса (активным накопителям согласно рис. 4.1, б), а переходы – межклассовым взаимодействиям. Пример ЭПД для процесса  $\Pi_0$  в потоковом сценарии компании по обслуживанию заказов приведен на рис. 4.13.

Классы диаграммы, так же как и соответствующие им операции сценария, изображаются прямоугольниками со скругленными углами. В прямоугольнике очеркивается левое и правое поле. В левое поле помещаются обозначения потоков входных объектов и ресурсов, в правое поле помещаются обозначения потоков выходных объектов.

В среднем поле прямоугольника класса помещаются таблица атрибутов этого класса и диаграмма жизненного цикла объектов класса. Эволюция объектов при реализации их жизненного цикла отображается в таблице атрибутов класса, а переходы в диаграмме жизненного цикла обусловлены текущим заполнением таб-

лицы атрибутов. Поэтому изображения таблицы и диаграммы «внутри класса» соединены двунаправленными стрелками.

Строки таблицы атрибутов класса  $\Pi_i$  соответствуют атрибутам  $a_{i,j}$ , а столбцы – объектам (экземплярам класса). В ходе реализации объектом жизненного цикла (внутриклассовой эволюции) изменяются значения неключевых атрибутов соответствующего столбца (т.е. объект отображается вектор-столбцом). В момент создания объекта в классе его вектор-столбец принимает начальные значения. Для наследуемых атрибутов начальные значения формируются в тех классах, из которых эти атрибуты наследуются. Начальные значения порождаемых атрибутов формируются при срабатывании переходов, соответствующих межклассовым взаимодействиям.

Диаграмма жизненного цикла объектов имеет вид ориентированного графа, вершины которого соответствуют состояниям объекта (фазам жизненного цикла), а дуги – переходам между состояниями. Вершины помечаются обозначением атрибутов, значения которых в данном состоянии изменяются, и даются правила этих изменений. Дуги помечаются условиями срабатывания переходов между состояниями – логическими функциями над атрибутами.

Переходы между классами изображаются планками (жирными черточками). Входные дуги перехода помечаются наследуемыми атрибутами, а выходные дуги – порождаемыми. Переходу приписываются условие срабатывания, выражаемое логической функцией от атрибутов входных/выходных классов, и действие, соответствующее присваиванию начальных значений порождаемым атрибутам.

ЭПД формируется на основе потокового сценария в следующем порядке.

1. Операции сценария объявляются классами.
2. Обозначения входных (выходных) объектных потоков операций заносятся в левое (правое) поле класса.
3. Ресурсы операций сценария, а возможно, и дополнительные ресурсы заносятся в левое поле класса.
4. Определяются атрибуты объектов каждого класса (структурирование объектов).
5. Строится диаграмма жизненного цикла для каждого класса.
6. Вводятся межклассовые переходы.
7. Из сценария переносятся на диаграмму горизонтальные стрелки, помеченные внешними потоками, направленные к наружной рамке и из наружной рамки.

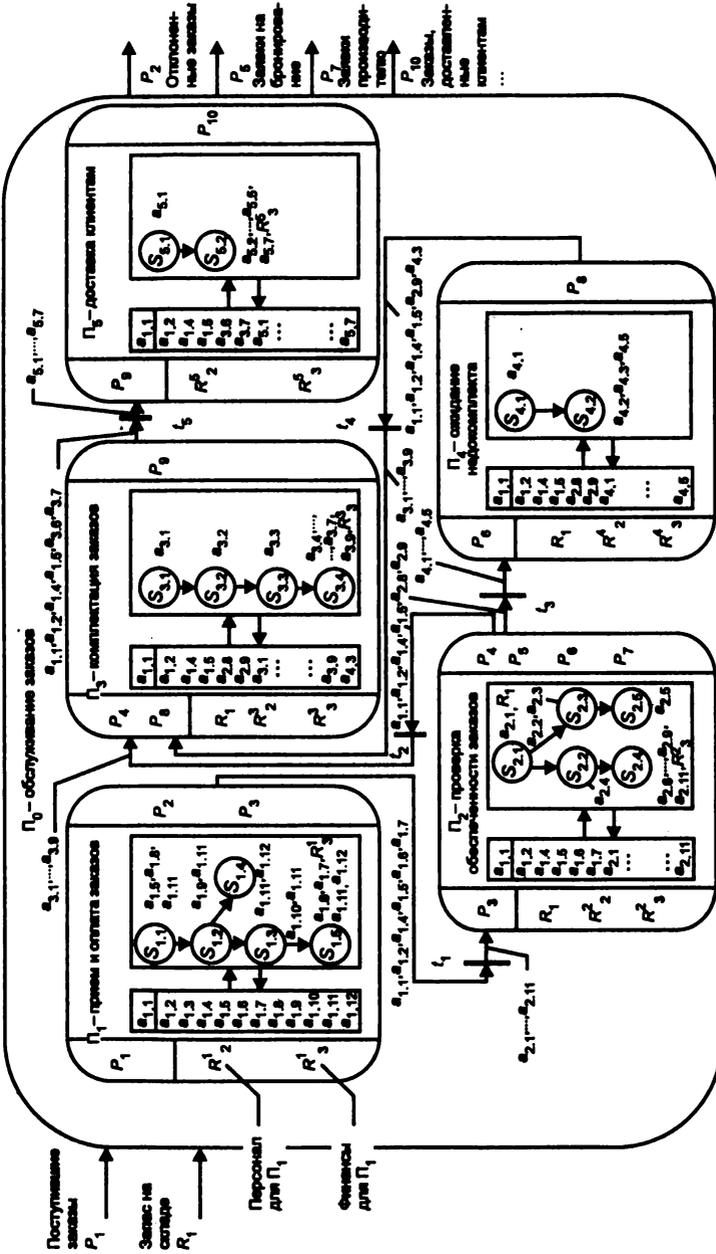


Рис. 4.13. Пример эволюционно-потоковой диаграммы (для системы обслуживания заказов)

Атрибуты объектов ЭПД (см. рис. 4.13)

Класс	Обозначение атрибута	Содержание атрибута
П <sub>1</sub>	<i>a</i> <sub>1.1</sub>	Идентификатор (ID) заказа
	<i>a</i> <sub>1.2</sub>	ID заказчика
	<i>a</i> <sub>1.3</sub>	Банковский счет заказчика
	<i>a</i> <sub>1.4</sub>	Сумма заказа
	<i>a</i> <sub>1.5</sub>	Дата поступления заказа
	<i>a</i> <sub>1.6</sub>	Время обработки заказа в классе П <sub>1</sub>
	<i>a</i> <sub>1.7</sub>	Затраты на обработку заказа в классе П <sub>1</sub>
	<i>a</i> <sub>1.8</sub>	Заказ зафиксирован
	<i>a</i> <sub>1.9</sub>	Заказ принят
	<i>a</i> <sub>1.10</sub>	Заказ оплачен
	<i>a</i> <sub>1.11</sub>	Состояние жизненного цикла заказа в П <sub>1</sub>
	<i>a</i> <sub>1.12</sub>	Обслуживание заказа в классе П <sub>1</sub> закончено
П <sub>2</sub>		<i>a</i> <sub>1.1</sub> , <i>a</i> <sub>1.2</sub> , <i>a</i> <sub>1.4</sub> , <i>a</i> <sub>1.5</sub> , <i>a</i> <sub>1.6</sub> , <i>a</i> <sub>1.7</sub> – атрибуты, наследуемые из П <sub>1</sub>
	<i>a</i> <sub>2.1</sub>	Заказ обеспечен запасом (ресурсом) <i>R</i> <sub>1</sub>
	<i>a</i> <sub>2.2</sub>	Недокомплект определен
	<i>a</i> <sub>2.3</sub>	Величина недокомплекта
	<i>a</i> <sub>2.4</sub>	Сформирована заявка на бронирование
	<i>a</i> <sub>2.5</sub>	Сформирована заявка производителю
	<i>a</i> <sub>2.6</sub>	Время обслуживания заказа в классе П <sub>2</sub>
	<i>a</i> <sub>2.7</sub>	Затраты на обслуживание заказа в классе П <sub>2</sub>
	<i>a</i> <sub>2.8</sub>	Суммарное время обслуживания заказа
	<i>a</i> <sub>2.9</sub>	Суммарные затраты на обслуживание заказа
	<i>a</i> <sub>2.10</sub>	Состояние жизненного цикла заказа в П <sub>2</sub>
<i>a</i> <sub>2.11</sub>	Обслуживание обеспеченного заказа в классе П <sub>2</sub> закончено	
П <sub>3</sub>		<i>a</i> <sub>1.1</sub> , <i>a</i> <sub>1.2</sub> , <i>a</i> <sub>1.4</sub> , <i>a</i> <sub>1.5</sub> , <i>a</i> <sub>2.8</sub> , <i>a</i> <sub>2.9</sub> – атрибуты, наследуемые из П <sub>2</sub> , <i>a</i> <sub>4.3</sub> – наследуемый из П <sub>4</sub>
	<i>a</i> <sub>3.1</sub>	Заказ собран
	<i>a</i> <sub>3.2</sub>	Заказ упакован
	<i>a</i> <sub>3.3</sub>	Сопроводительная документация оформлена
	<i>a</i> <sub>3.4</sub>	Время обслуживания заказа в классе П <sub>3</sub>
	<i>a</i> <sub>3.5</sub>	Затраты на обслуживание заказа в классе П <sub>3</sub>

Класс	Обозначение атрибута	Содержание атрибута
	$a_{3,6}$ $a_{3,7}$ $a_{3,8}$ $a_{3,9}$	Суммарное время обслуживания заказа Суммарные затраты на обслуживание заказа Состояние жизненного цикла заказа в $\Pi_3$ Обслуживание заказа в классе $\Pi_3$ закончено
$\Pi_4$	$a_{4,1}$ $a_{4,2}$ $a_{4,3}$ $a_{4,4}$ $a_{4,5}$	$a_{1,1}, a_{1,2}, a_{1,4}, a_{1,5}, a_{2,8}, a_{2,9}$ – атрибуты, наследуемые из $\Pi_2$ Недокомплект поступил на склад Время ожидания заказа в классе $\Pi_4$ Суммарное время обслуживания заказа Состояние жизненного цикла заказа в $\Pi_4$ Заказ передан из $\Pi_4$ на комплектацию
$\Pi_5$	$a_{5,1}$ $a_{5,2}$ $a_{5,3}$ $a_{5,4}$ $a_{5,5}$ $a_{5,6}$ $a_{5,7}$	$a_{1,1}, a_{1,2}, a_{1,4}, a_{1,5}, a_{3,6}, a_{3,7}$ – атрибуты, наследуемые из $\Pi_3$ Заказ доставлен клиенту Время операции доставки Затраты на доставку Суммарное время обслуживания заказа Суммарные затраты на обслуживание заказа Состояние жизненного цикла заказа в $\Pi_5$ Заказ «погашен» (обслуживание закончено)

Вернемся к примеру, показанному на рис. 4.13. Состав и содержание атрибутов всех классов даны в табл. 4.3. Ключевым во всех классах является атрибут  $a_{1,1}$  «Идентификатор заказа», определяющий входные объекты  $P_1$  класса  $\Pi_1$  и наследуемый во все остальные классы (при изображении классов отделен горизонтальной чертой от остальных атрибутов).

В качестве ресурсов, помимо запаса  $R_1$  на складе, используемого в классах  $\Pi_2, \Pi_3, \Pi_4$ , в каждом классе  $\Pi_i, i = 1, \dots, 5$ , используются ресурсы  $R_2^i, R_3^i$ , соответствующие персоналу и финансовому обеспечению. Заметим, что ресурсы чаще всего выражают в денежном исчислении. Сложные ресурсы структурируют, т.е. выделяют их атрибуты.

Комментарий к описанию жизненного цикла объектов в ЭПД будет дан в разд. 4.4.1, к описанию внутрикласовых и межкласовых преобразований – в разд. 4.4.2. В разд. 4.4.3 рассмотрим

некоторые интегральные характеристики функционирования системы, получаемые в результате «прогона» потокового сценария, и оценочные зависимости для определения степени достижимости поставленных целей, в разд. 4.4.4 подробно обсудим механизм функционирования активной потоковой модели, в разд. 4.4.5 обобщим понятие «конвейерно-временная диаграмма».

#### 4.4.1.

#### Описание жизненного цикла объектов для классов эволюционно-потоковой диаграммы

Текстовое описание состояний жизненного цикла объектов для всех классов ЭПД на рис. 4.13 дано в табл. 4.4, а условия перехода между состояниями – в табл. 4.5. При формировании условий применены значения атрибутов ДА, НЕТ (в смысле ИСТИНА, ЛОЖЬ).

Таблица 4.4

Состояние жизненного цикла объектов (см. рис. 4.13)

Класс	Обозначение состояния	Содержание состояния
П <sub>1</sub>	S <sub>1,1</sub>	Фиксирование поступления заказа
	S <sub>1,2</sub>	Проверка платежеспособности клиента
	S <sub>1,3</sub>	Оплата заказа
	S <sub>1,4</sub>	Передача отклоненного заказа в выходной накопитель P <sub>2</sub>
	S <sub>1,5</sub>	Документирование принятого заказа и передача его в выходной накопитель P <sub>3</sub>
П <sub>2</sub>	S <sub>2,1</sub>	Проверка обеспеченности заказа ресурсом R <sub>1</sub>
	S <sub>2,2</sub>	Бронирование заказа
	S <sub>2,3</sub>	Расчет величины недокомплекта
	S <sub>2,4</sub>	Определение времени, затрат и остаточного ресурса R <sub>3</sub> <sup>2</sup> при обслуживании заказа
	S <sub>2,5</sub>	Формирование заявки производителю
П <sub>3</sub>	S <sub>3,1</sub>	Сборка компонентов заказа
	S <sub>3,2</sub>	Упаковка заказа
	S <sub>3,3</sub>	Оформление сопроводительной документации
	S <sub>3,4</sub>	Определение времени, затрат и остаточного ресурса R <sub>3</sub> <sup>3</sup> при обслуживании заказа

Класс	Обозначение состояния	Содержание состояния
$\Pi_4$	$S_{4,1}$ $S_{4,2}$	Ожидание поступления недокомплекта Определение времени ожидания и суммарного времени обслуживания заказа
$\Pi_5$	$S_{5,1}$ $S_{5,2}$	Доставка заказа Определение времени, затрат и остаточного ресурса $R_3^5$ при доставке заказа, а также суммарных времени и затрат на «сквозное» обслуживание заказа

Таблица 4.5

**Условия перехода между состояниями жизненного цикла объектов**  
(см. рис. 4.13)

Класс	Обозначение перехода	Условие перехода
$\Pi_1$	$S_{1,1}S_{1,2}$	$a_{1,8} = \text{ДА}$
	$S_{1,2}S_{1,3}$	$a_{1,9} = \text{ДА}$
	$S_{1,2}S_{1,4}$	$a_{1,9} = \text{НЕТ}$
	$S_{1,3}S_{1,5}$	$a_{1,10} = \text{ДА}$
$\Pi_2$	$S_{2,1}S_{2,2}$	$a_{2,1} = \text{ДА}$
	$S_{2,1}S_{2,3}$	$a_{2,1} = \text{НЕТ}$
	$S_{2,2}S_{2,4}$	$a_{2,4} = \text{ДА}$
	$S_{2,3}S_{2,5}$	$a_{2,2} = \text{ДА}$
$\Pi_3$	$S_{3,1}S_{3,2}$	$a_{3,1} = \text{ДА}$
	$S_{3,2}S_{3,3}$	$a_{3,2} = \text{ДА}$
	$S_{3,3}S_{3,4}$	$a_{3,3} = \text{ДА}$
$\Pi_4$	$S_{4,1}S_{4,2}$	$a_{4,1} = \text{ДА}$
$\Pi_5$	$S_{5,1}S_{5,2}$	$a_{5,1} = \text{ДА}$

В качестве примера рассмотрим диаграмму жизненного цикла для класса  $\Pi_1$  (см. рис. 4.13). Она имеет начальное состояние  $S_{1,1}$ , в котором фиксируется поступление на обслуживание каждого объекта-заказа  $P_{1,i}$  из входного накопителя  $P_1$ , и два конечных состояния  $S_{1,4}$ ,  $S_{1,5}$ . Состояние  $S_{1,4}$  свидетельствует о том, что заказ отклонен, состояние  $S_{1,5}$  – что заказ принят к исполнению. Принятые заказы направляются в выходной накопитель  $P_3$ , отклоненные – в выходной накопитель  $P_2$ .

Будем считать, что в заказах из входного накопителя  $P_1$  заполнены атрибуты  $a_{1,1}, \dots, a_{1,4}$  (идентификатор заказа, идентификатор и номер банковского счета заказчика, сумма заказа), а остальные атрибуты по умолчанию имеют нулевое значение (для логических атрибутов – значение НЕТ).

В начальном состоянии присваивается значение атрибуту  $a_{1,5}$  «Дата поступления заказа на обслуживание» и атрибуту  $a_{1,8}$ , фиксирующему факт поступления заказа. Атрибут  $a_{1,8}$  является логической переменной, и ему присваивается значение ДА. Кроме того, «означивается» атрибут  $a_{1,11}$ , указывающий то состояние жизненного цикла, в котором в настоящий момент находится заказ. В начальном состоянии  $a_{1,11} = S_{1,1}$ . На диаграмме жизненного цикла объектов в классе  $\Pi_1$  (см. рис. 4.13) рядом с кружком состояния  $S_{1,1}$  проставлены атрибуты  $a_{1,5}, a_{1,8}, a_{1,11}$ , «означиваемые» в начальном состоянии (для остальных классов ЭПД на рис. 4.13 атрибут «Состояние жизненного цикла», которым должна была бы помечаться каждая вершина соответствующей диаграммы, по умолчанию опущен).

Переход из  $S_{1,1}$  в состояние  $S_{1,2}$ , в котором проверяется платежеспособность заказчика, происходит при выполнении условия  $a_{1,8} = \text{ДА}$  (см. табл. 4.5). В состоянии  $S_{1,2}$  логическому атрибуту  $a_{1,9}$  «Заказ принят» в зависимости от результата проверки присваивается значение ДА или НЕТ (а атрибуту  $a_{1,11}$  – значение  $S_{1,2}$ ). По условию  $a_{1,9} = \text{ДА}$  заказ переходит в состояние  $S_{1,3}$  «Оплата заказа», а по условию  $a_{1,9} = \text{НЕТ}$  – в конечное состояние  $S_{1,4}$ , из которого он передается в выходной накопитель  $P_2$ . В состоянии  $S_{1,3}$  помимо «означивания»  $a_{1,11} = S_{1,3}$  фиксируется факт оплаты заказа ( $a_{1,10} = \text{ДА}$ ), и по этому условию осуществляется переход в конечное состояние  $S_{1,5}$ , в котором «означиваются» атрибуты  $a_{1,6}, a_{1,7}$  (время и финансовые затраты на обработку заказа в классе  $\Pi_1$ ), определяется остаточный ресурс  $R_3$ , фиксируется факт завершения обслуживания заказа в классе  $\Pi_1$  ( $a_{1,12} = \text{ДА}$ ), выполняется  $a_{1,11} = S_{1,5}$  и осуществляется передача документированного заказа в выходной накопитель  $P_3$ .

Таким образом, прохождение каждым заказом своего жизненного цикла сопровождается изменением значений неключевых атрибутов в том столбце «внутренней» таблицы класса, который соответствует данному заказу, а также, возможно, изменением состояния ресурсов. Указанные изменения будем называть внутриклассовыми преобразованиями (в отличие от межклассовых

преобразований, которые связаны с уничтожением («смертью») объектов в одних классах и их созданием («рождением») в других). Примеры описания внутриклассовых и межклассовых преобразований даны в разд. 4.4.2.

В заключение отметим, что для каждого объекта характерна определенная «история жизни» в классе, которая описывается последовательностью наборов значений его атрибутов, «привязанной» к шкале реального времени. «Истории жизни» объектов (в классах) определяются в результате имитационного моделирования ЭПД и могут быть весьма полезными при принятии решений.

#### 4.4.2.

#### Описание внутриклассовых и межклассовых преобразований в активной потоковой модели

Правила *внутриклассовых преобразований* для ЭПД на рис. 4.13 даны в табл. 4.6. Среди них будем различать преобразования, отнесенные к определенному состоянию жизненного цикла объектов, и общие преобразования, которые могут выполняться в любой момент жизненного цикла. Последние будем называть блокировками.

Таблица 4.6

Внутриклассовые преобразования (см. рис. 4.13)

Класс	Состояние жизненного цикла	Преобразования
П <sub>1</sub>	S <sub>1,1</sub>	a <sub>1,5</sub> = <год, мес., число, текущее время> a <sub>1,8</sub> = ДА
	S <sub>1,2</sub>	Если <заказчик платежеспособен и документация на заказ отвечает принятым нормам>, то a <sub>1,9</sub> = ДА, иначе a <sub>1,9</sub> = НЕТ.
	S <sub>1,3</sub>	Если <выполнено перечисление либо оплата наличными >, то a <sub>1,10</sub> = ДА, иначе a <sub>1,10</sub> = НЕТ.
	S <sub>1,4</sub>	a <sub>1,12</sub> = ДА
	S <sub>1,5</sub>	a <sub>1,6</sub> = V <sub>1,i</sub> , где V <sub>1,i</sub> – время обслуживания в классе П <sub>1</sub> i-го заказа, i = 1,2,...; a <sub>1,7</sub> = Z <sub>1,i</sub> , где Z <sub>1,i</sub> – затраты на i-й заказ; R <sub>3</sub> <sup>1</sup> = R <sub>3</sub> <sup>1</sup> - Z <sub>1,i</sub> ; a <sub>1,12</sub> = ДА.

Класс	Состояние жизненного цикла	Преобразования
	<p>Ресурсная блокировка: Если <math>(R_2^1 &lt; R_{2\min}^1) \vee (R_3^1 &lt; R_{3\min}^1)</math>, то &lt;остановить работу класса П<sub>1</sub>&gt;</p>	
П <sub>2</sub>	S <sub>2.1</sub>	<p>Если <math>a_{1,4} \leq R_1</math>, то <math>a_{2,1} = \text{ДА}</math>, иначе <math>a_{2,1} = \text{НЕТ}</math>.</p>
	S <sub>2.2</sub>	<p>Если <math>V_{2,2i} \geq V_{2,2}^N</math>, то <math>a_{2,4} = \text{ДА}</math>, иначе <math>a_{2,4} = \text{НЕТ}</math>, <math>V_{2,2,i}</math> – текущее время обслуживания <math>i</math>-го заказа в состоянии S<sub>2.2</sub>, <math>V_{2,2}^N</math> – норма времени обслуживания в S<sub>2.2</sub>.</p>
	S <sub>2.3</sub>	<p>Если <math>Z_{2,i} &gt; R_1</math>, то <math>a_{2,2} = \text{ДА}</math>, иначе <math>a_{2,2} = \text{НЕТ}</math>; <math>a_{2,3} = a_{1,4} - R_1</math></p>
	S <sub>2.4</sub>	<p><math>a_{2,6} = V_{2,i}</math>, <math>a_{2,7} = Z_{2,i}</math>, <math>a_{2,8} = a_{1,6} + V_{2,i}</math>, <math>a_{2,9} = a_{1,7} + Z_{2,i}</math>, <math>R_3^2 = R_3^2 - Z_{2,i}</math></p>
	S <sub>2.5</sub>	<p>Если <math>V_{2,5i} \geq V_{2,5}^N</math>, то <math>a_{2,5} = \text{ДА}</math>, иначе <math>a_{2,5} = \text{НЕТ}</math>; <math>a_{2,11} = \text{ДА}</math></p>
		<p>Ресурсная блокировка: Если <math>(R_1 &lt; R_{1\min}) \vee (R_2^2 &lt; R_{2\min}^2) \vee (R_3^2 &lt; R_{3\min}^2)</math>, то &lt;остановить работу класса П<sub>2</sub>&gt;</p>
П <sub>3</sub>	S <sub>3.1</sub>	<p>Если <math>V_{3,1i} \geq V_{3,1}^N</math>, то <math>a_{3,1} = \text{ДА}</math>, иначе <math>a_{3,1} = \text{НЕТ}</math></p>
	S <sub>3.2</sub>	<p>Если <math>V_{3,2i} \geq V_{3,2}^N</math>, то <math>a_{3,2} = \text{ДА}</math>, иначе <math>a_{3,2} = \text{НЕТ}</math></p>
	S <sub>3.3</sub>	<p>Если <math>V_{3,3i} \geq V_{3,3}^N</math>, то <math>a_{3,3} = \text{ДА}</math>, иначе <math>a_{3,3} = \text{НЕТ}</math></p>

Класс	Состояние жизненного цикла	Преобразования
	$S_{3,4}$	$a_{3,4} = V_{3,l}$ $a_{3,5} = Z_{3,l}$ $a_{3,6} = a_{2,8} + V_{3,l}$ $a_{3,7} = a_{2,9} + Z_{3,l}$ $a_{3,9} = \text{ДА}$ $R_1 = R_1 - a_{1,4}, R_3^3 = R_3^3 - Z_{3,l}$
	<p>Ресурсная блокировка:</p> <p>Если <math>(R_1 &lt; R_{1 \min}) \vee (R_2^3 &lt; R_{2 \min}^3) \vee (R_3^3 &lt; R_{3 \min}^3)</math>,  то &lt;остановить работу класса <math>\Pi_3</math>&gt;</p>	
$\Pi_4$	$S_{4,1}$	<p>Если <math>a_{1,4} \leq R_1</math>,  то <math>a_{4,1} = \text{ДА}</math>,  иначе <math>a_{4,1} = \text{НЕТ}</math></p>
	$S_{4,2}$	$a_{4,2} = V_{4,l}$ $a_{4,3} = a_{2,8} + V_{4,l}$ $a_{4,5} = \text{ДА}$
	<p>Ресурсная блокировка:</p> <p>Если <math>(R_1 &lt; R_{1 \min}) \vee (R_2^4 &lt; R_{2 \min}^4) \vee (R_3^4 &lt; R_{3 \min}^4)</math>,  то &lt;остановить работу класса <math>\Pi_4</math>&gt;</p>	
$\Pi_5$	$S_{5,1}$	<p>Если <math>V_{5,1l} \geq V_{5,1}^N</math>,  то <math>a_{5,1} = \text{ДА}</math>,  иначе <math>a_{5,1} = \text{НЕТ}</math></p>
	$S_{5,2}$	$a_{5,2} = V_{5,l}$ $a_{5,3} = Z_{5,l}$ $a_{5,4} = a_{3,6} + V_{5,l}$ $a_{5,5} = a_{3,7} + Z_{5,l}$ $a_{5,7} = \text{ДА}$ $R_3^5 = R_3^5 - Z_{5,l}$
	<p>Ресурсная блокировка:</p> <p>Если <math>(R_2^5 &lt; R_{2 \min}^5) \vee (R_3^5 &lt; R_{3 \min}^5)</math>,  то &lt;остановить работу класса <math>\Pi_5</math>&gt;</p>	

Преобразования в состояниях жизненного цикла в общем случае проводятся по схеме:

Если <выполнение условия, выраженного логической переменной>,

то <действие 1 по «означиванию» атрибутов>,  
иначе <действие 2 по «означиванию» атрибутов>.

В частном случае условие может отсутствовать и преобразование ограничивается одним действием.

Преобразования-блокировки, как правило, определяются ресурсами: при уменьшении количества ресурсов ниже некоторого уровня реализация жизненного цикла объектов прерывается (в текущем состоянии), при пополнении ресурсов – возобновляется из текущего состояния.

Согласно табл. 4.6 в начальном состоянии  $S_{1,1}$  класса  $\Pi_1$  проводятся безусловные действия, прокомментированные в предыдущем разделе (действия по «означиванию» атрибутов, соответствующих состояниям жизненного цикла объектов, в табл. 4.6 по умолчанию опускаем). В состоянии  $S_{1,2}$ , как уже говорилось, проверяются платежеспособность заказчика и соответствие представленной им документации принятым нормам в зависимости от результата проверки атрибуту  $a_{1,9}$  присваивается значение ДА или НЕТ. Таким образом, преобразование в состоянии  $S_{1,2}$  соответствует схеме Если–то–иначе. По аналогичной схеме выполняется преобразование и в состоянии  $S_{1,3}$ . В состояниях  $S_{1,4}$ ,  $S_{1,5}$  выполняются безусловные действия. При этом в  $S_{1,5}$  атрибутам  $a_{1,6}$ ,  $a_{1,7}$  присваиваются соответственно показания таймера  $V_{1,i}$ , зафиксировавшего время обслуживания  $i$ -го заказа в классе  $\Pi_1$ , и расчетные затраты  $Z_{1,i}$  на  $i$ -й заказ. Остаточный финансовый ресурс (после обработки  $i$ -го заказа) вычисляется по формуле  $R_3^1 = R_3^1 - Z_{1,i}$ .

Ресурсная блокировка срабатывает, если число сотрудников, занятых обслуживанием заказов, или финансирование системы снижается ниже критического уровня.

Подобным образом в табл. 4.6 описаны внутренние преобразования для остальных классов ЭПД по рис. 4.13.

*Межклассовые взаимодействия* моделируются переходами сети Петри (см. рис. 4.13), входные дуги которых помечены множеством атрибутов, наследуемых из входного класса перехода в его выходные классы, а выходные дуги перехода – множеством атрибутов, порождаемых в выходном классе. Если класс имеет несколько входных переходов, то порождаемые ими множества атрибутов совпадают.

Множество атрибутов класса образуется объединением указанных порождаемых атрибутов и всех наследуемых атрибутов (из классов – родителей данного класса). Наследуемые атрибуты попадают в класс с теми значениями, которые они получили в родительских классах, а начальные значения порождаемых атрибутов формируются при срабатывании входных переходов класса. При этом должны выполняться два условия:

- если атрибут наследуется из нескольких родительских классов, он должен получать в них одинаковое значение;
- если класс имеет несколько входных переходов, то каждый из них присваивает каждому атрибуту одно и то же начальное значение.

Переходу между классами приписывается выражение: Если <выполнение логического условия, разрешающего запуск перехода>, то <действия по уничтожению «отживших» объектов во входных (родительских) классах и порождению новых объектов в выходных классах (потомках)>. Действия сопровождаются наследованием значений атрибутов, указанных на входных дугах перехода, и начальным «означиванием» атрибутов, указанных на его выходных дугах (с соблюдением вышеприведенных условий).

Запуск переходов между классами определяется следующими соображениями. Примем, что очереди объектов в накопителе на входе (выходе) классов имеют фиксированную максимальную длину. Переход запускается, если:

- в выходных очередях его родительских классов есть по меньшей мере один объект;
- во входных очередях его классов-потомков есть «свободное место»;
- выполняется условие запуска, выражаемое логической функцией над состояниями жизненного цикла и атрибутами входных (выходных) классов перехода.

В ЭПД на рис. 4.13 переход  $t_1$  с входным классом  $\Pi_1$  и выходным  $\Pi_2$  наследует атрибуты  $a_{1,1}, a_{1,2}, a_{1,4}, a_{1,5}, a_{1,6}, a_{1,7}$  и порождает атрибуты  $a_{2,1}, \dots, a_{2,11}$ . Их объединение определяет состав атрибутов класса  $\Pi_2$ . Логическое условие запуска  $t_1$  для примера на рис. 4.13 говорит о том, что уничтожаемый в классе  $\Pi_1$  объект должен находиться в конечном состоянии ( $S_{1,5} = \text{ДА}$ ) и его обслуживание должно быть закончено ( $a_{1,12} = \text{ДА}$ ). Класс  $\Pi_3$  имеет два входных перехода –  $t_2$  из класса  $\Pi_2$  и  $t_4$  из класса  $\Pi_4$ . Они наследуют совпадающие атрибуты  $a_{1,1}, a_{1,2}, a_{1,4}, a_{1,5}, a_{2,9}$  (которые «по-

кидают» классы  $\Pi_2$  и  $\Pi_4$  с одинаковыми значениями) и несовпадающие атрибуты  $a_{2,8}$ ,  $a_{4,3}$  и порождают атрибуты  $a_{3,1}, \dots, a_{3,9}$ . Все наследуемые и порожденные атрибуты входят в класс  $\Pi_3$ . (Напомним, что при «означивании» порождаемых атрибутов в результате срабатывания переходов между классами числовым атрибутам по умолчанию присваивается нулевое значение, а логическим атрибутам – значение НЕТ). Условие запуска переходов  $t_2, \dots, t_5$  подобно условию для  $t_1$ . В соответствии с рис. 4.13 ниже приведены условия запуска переходов между классами.

Переход	Условие
$t_1$	$(S_{1,5} = \text{ДА}) \wedge (a_{1,12} = \text{ДА})$
$t_2$	$(S_{2,4} = \text{ДА}) \wedge (a_{2,11} = \text{ДА})$
$t_3$	$(S_{2,5} = \text{ДА}) \wedge (a_{2,5} = \text{ДА})$
$t_4$	$(S_{4,2} = \text{ДА}) \wedge (a_{4,5} = \text{ДА})$
$t_5$	$(S_{3,4} = \text{ДА}) \wedge (a_{3,9} = \text{ДА})$

Итак, укажем, что если объекты создаются, перемещаются (мигрируют) и уничтожаются внутри классов, то их атрибуты могут мигрировать (с помощью механизма наследования) и между классами. Поскольку атрибуты являются носителями свойств элементов потоков, такая межклассовая миграция помогает лучше понять и оценить поведение системы.

### 4.4.3.

#### Оценка достижимости целей по результатам моделирования эволюционно-потокковой диаграммы

В результате имитационного моделирования («прогона») ЭПД определяются не только «истории жизни» объектов (при внутриклассовой миграции) и атрибутов (при межклассовой миграции), но и интегральные показатели поведения организационной системы в целом, отнесенные к заданному интервалу времени. Приведем примеры интегральных показателей  $I_d$  для ЭПД на рис. 4.13.

Обозначение	Содержание
$I_1$	Годовой (квартальный, месячный) объем реализации заказов (в у.е.)

Обозначение	Содержание
$I_2$	Годовые (квартальные, месячные) затраты на обслуживание заказов (в у.е.)
$I_3$	Среднее время обслуживания заказа (в днях) в течение года (квартала, месяца)
$I_4$	Средние затраты на обслуживание заказа (в у.е.) в течение года (квартала, месяца) и т.д.

На основе интегральных показателей и факторов влияния внешней среды можно грубо оценить степень достижимости целей системы.

Обозначение	Содержание
$\Phi_1$	Размещение рекламы на телевидении, радио, в периодической печати и т.д.
$\Phi_2$	Размещение рекламы в Интернете
$\Phi_3$	Взаимодействие с клиентом по принципу интернет-магазина
$\Phi_4$	Возможность по желанию клиента создания новых товаров
$\Phi_5$	Рыночная репутация корпорации
$\Phi_6$	Потенциал корпорации (финансовое состояние, состояние фондов и т.д.)
$\Phi_7$	Пересылка заявок производителю по мере поступления заказов (производитель осуществляет непрерывное планирование производства и, возможно, проектирования)
$\Phi_8$	Использование наряду с автотранспортом авиатранспорта
$\Phi_9$	Сокращение избыточного персонала
$\Phi_{10}$	Повышение квалификации персонала (обучение)
$\Phi_{11}$	Повышение зарплаты и улучшение социальных условий для персонала
$\Phi_{12}$	Развертывание в системе обслуживания заказов локальной компьютерной сети

Достижимость целей  $C_i$  и факторы  $\Phi_j$  трактуются как лингвистические переменные [3], изменяющиеся на  $k$ -балльной (например, 5-балльной) шкале, максимальная и минимальная границы которой соответственно  $k$  и 1. Каждое значение степени достижимости цели  $C_i$  выражается логической функцией  $L(I_\Phi, \Phi_j)$ :

$$C_i = X, \text{ если } L(I_\Phi, \Phi_j) = \text{ДА},$$

где  $X = 1, \dots, k$ .

Пример представления такой зависимости для ЭПД на рис. 4.13 приведен в табл. 4.7. Степень достижимости цели – «Увеличить объем заказов»  $C_1=5$ :

Если <годовой объем реализации заказов не менее 1 млн. у.е.>  
 $\wedge$  <годовые затраты на обслуживание заказов не превышают 15% от объема реализации>

$\wedge$  <размещение в Интернете рекламы услуг оценивается как хорошее или отличное:  $\Phi_2 \geq 4$ >

$\wedge$  <повышение квалификации персонала в компании проводится хорошо или отлично:  $\Phi_{10} \geq 4$ >

$\wedge$  <повышение зарплаты и улучшение социальных условий персонала оценивается как хорошее или отличное:  $\Phi_{11} \geq 4$ >.

Подобным образом составлены и остальные выражения табл. 4.7.

Таблица 4.7

Зависимость степени достижимости целей системы от интегральных показателей поведения  $I$  и факторов влияния  $\Phi$

Цели			Зависимость от $I$ и $\Phi$
Обозначение	Содержание	Степень достижимости (в баллах)	
$C_1$	Увеличить объем заказов	5	$(I_1 \geq 1 \text{ млн}) \wedge (I_2 \leq 0,15 I_1) \wedge (\Phi_2 \geq 4) \wedge (\Phi_{10} \geq 4) \wedge (\Phi_{11} \geq 4)$ .
		4	$(I_1 \geq 800 \text{ тыс.}) \wedge (I_2 \leq 0,2 I_1) \wedge (\Phi_2 \geq 4) \wedge (\Phi_{10} \geq 4) \wedge (\Phi_{11} \geq 4)$ .
		3	$(I_1 \geq 600 \text{ тыс.}) \wedge (I_2 \leq 0,25 I_1) \wedge (\Phi_2 = 3) \wedge (\Phi_{10} = 3) \wedge (\Phi_{11} = 3)$ .
		2	$(I_1 \geq 500 \text{ тыс.}) \wedge (I_2 \leq 0,3 I_1) \wedge (\Phi_2 = 2) \wedge (\Phi_{10} = 2) \wedge (\Phi_{11} = 2)$ .
		1	$(I_1 \geq 400 \text{ тыс.}) \wedge (I_2 \leq 0,3 I_1) \wedge (\Phi_2 \leq 2) \wedge (\Phi_{10} \leq 2) \wedge (\Phi_{11} \leq 2)$ .
$C_2$	Сократить время обслуживания заказов	5	$(I_3 \leq 2) \wedge (\Phi_{10} = 5) \wedge (\Phi_{11} \geq 4)$
		4	$(I_3 \leq 3) \wedge (\Phi_{10} \geq 4) \wedge (\Phi_{11} \geq 4)$
		3	$(I_3 \leq 4) \wedge (\Phi_{10} = 3) \wedge (\Phi_{11} = 3)$
		2	$(I_3 \leq 10) \wedge (\Phi_{10} = 3) \wedge (\Phi_{11} = 3)$
		1	$(I_3 \leq 15) \wedge (\Phi_{10} \leq 2) \wedge (\Phi_{11} \leq 2)$
$C_3$	Поднять уровень организации труда	5	$(I_4 \leq 10) \wedge (\Phi_7 = 5) \wedge (\Phi_{12} = 5)$
		4	$(I_4 \leq 15) \wedge (\Phi_7 \geq 4) \wedge (\Phi_{12} = 5)$
		3	$(I_4 \leq 20) \wedge (\Phi_7 = 3) \wedge (\Phi_{12} \geq 4)$
		2	$(I_4 \leq 25) \wedge (\Phi_7 = 2) \wedge (\Phi_{12} = 3)$
		1	$(I_4 \leq 30) \wedge (\Phi_7 \leq 2) \wedge (\Phi_{12} \leq 2)$

Выражения  $L(I_d, \Phi_j)$ , естественно, субъективны и отражают точку зрения эксперта или группы экспертов.

#### 4.4.4.

#### **Механизм функционирования динамической модели, реализующей эволюционно-потокową диаграмму**

Выше при описании ЭПД мы коснулись некоторых аспектов функционирования реализующей ее динамической модели. Здесь же рассмотрим развернутый механизм функционирования этой модели.

Каждый класс ЭПД (см. рис. 4.13) работает автономно. Его взаимодействие с «внешним миром», т.е. с другими классами и внешней средой, заключается во внесении во входные очереди новых объектов и удалении из выходных очередей «отработавших» объектов. Функционирование класса обеспечивается потреблением необходимых ресурсов.

Емкость (максимальное число содержащихся объектов) вход-выходных очередей класса, а также вершин диаграммы его жизненного цикла, естественно, ограничена. Примем, что в очереди может содержаться не более  $n$  объектов, а в вершине диаграммы – не более  $h$ . Одновременно в классе свой жизненный цикл реализуют не более  $m = kh$  объектов, где  $k$  – число вершин диаграммы. Позициям (местам) очереди в порядке от «хвоста» к «голове» присваиваются номера  $1, 2, \dots, n$ .

Механизм функционирования класса (на примере фрагмента ЭПД на рис. 4.13) будем иллюстрировать схемой, показанной на рис. 4.14. Очереди объектов изображены таблицами, строки которых соответствуют атрибутам объектов, а столбцы – местам очереди. Каждый столбец отображает структуру объекта, находящегося на соответствующем месте.

Каждому объекту, который находится в вершине диаграммы жизненного цикла, соответствует свой столбец во «внутренней» таблице (на рис. 4.13 таблица соединена с диаграммой двумя противоположно направленными стрелками, что отражает взаимодействие диаграммы и внутренней таблицы). По мере прохождения объектом жизненного цикла изменяются значения атрибутов, вписанные в соответствующий этому объекту столбец внутренней таблицы, а переходы объекта из вершины в вершину зависят от текущего состояния данного столбца. Поскольку общее число

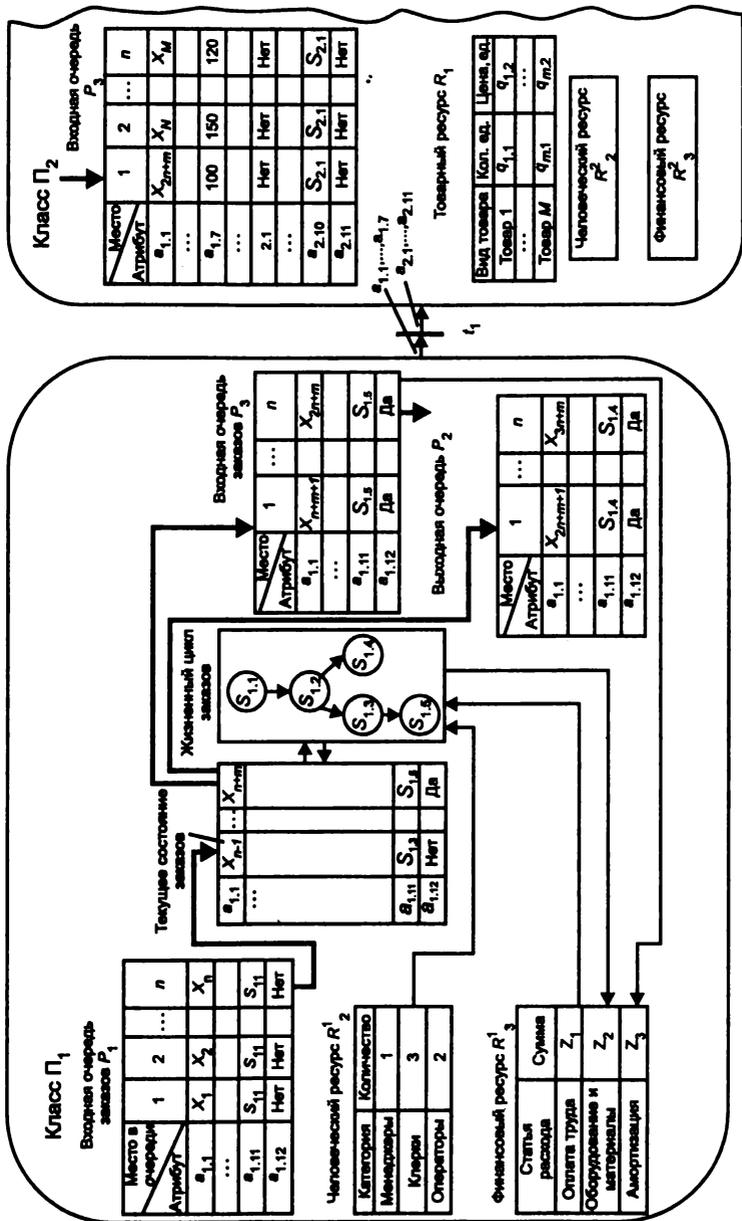


Рис. 4.14. Механизм функционирования динамической модели, реализующей ЭПД

объектов, присутствующих на диаграмме жизненного цикла, изменяется во времени, то изменяется и число столбцов внутренней таблицы.

Ресурсы класса также представляются таблицами, где строки – атрибуты ресурсов, а в столбцы вписаны текущие значения атрибутов.

Функционирование класса происходит следующим образом. Объекты, находящиеся во входной очереди, адресованы в начальную вершину диаграммы жизненного цикла (на рис. 4.14 во всех столбцах таблицы входной очереди заказов  $P_1$   $a_{1,11} = S_{1,1}$ ). Когда в начальной вершине появляется свободное место, т.е. число объектов в ней меньше максимального  $h$ , столбец входной очереди с наибольшим номером удаляется из очереди и вносится во внутреннюю таблицу (на рис. 4.14 этот перенос показан жирной стрелкой). Очередь сдвигается на шаг вправо, а объект, внесенный во внутреннюю таблицу, начинает реализацию своего жизненного цикла.

Когда какой-нибудь объект завершит жизненный цикл (на рис. 4.14 – заказ с идентификатором  $X_{n+m}$ , для которого  $a_{1,11} = S_{1,5}$ ,  $a_{1,12} = \text{ДА}$ ), соответствующий столбец удаляется из внутренней таблицы и вносится в выходную очередь, определяемую значением его атрибутов, на место с номером 1 (на рис. 4.14 это также показано жирными стрелками). Наличие двух стрелок обусловлено тем, что из конечной вершины  $S_{1,5}$  заказ передается в выходную очередь  $P_3$ , а из  $S_{1,4}$  – в  $P_2$ , выбор очереди определяется значением атрибута  $a_{1,11}$ ).

При срабатывании внешних переходов, для которых данный класс является родительским, из выходной очереди удаляется объект с наибольшим номером, а во входную очередь класса-потомка вносится новый объект (на место с номером 1). Атрибуты класса-потомка и их начальные значения определяются так, как это было показано в разд. 4.4.2. В схеме на рис. 4.14 из выходной очереди  $P_3$  класса  $\Pi_1$  удаляется заказ с идентификатором  $X_{2n+m}$ , а во входную очередь  $P_3$  класса  $\Pi_2$  вносится заказ, наследующий тот же идентификатор (показано жирными стрелками).

Таблицы, отображающие ресурсы класса, взаимодействуют с диаграммой жизненного цикла: в вершинах диаграммы могут выполняться действия, приводящие к расходованию ресурсов, а при критическом снижении уровня ресурсов может блокироваться срабатывание переходов диаграммы. Действия по расходованию

ресурсов можно выполнять не в вершинах диаграммы жизненного цикла, а при удалении объекта из выходной очереди (списывать расходы на объект при его уничтожении).

В примере на рис. 4.14 указанные взаимодействия показаны тонкими стрелками. Человеческий ресурс  $R_2^1$  только блокирует переходы (одностороннее взаимодействие); финансовый ресурс  $R_3^1$  как блокирует переходы, так и расходуется при выполнении жизненного цикла заказов или/и удалении заказов из выходной очереди (двустороннее взаимодействие). Аналогично одностороннее взаимодействие имеет место для товарного ресурса  $R_1$  в классе  $\Pi_3$  и двустороннее для него же в классе  $\Pi_2$  (см. рис. 4.13, 4.14) и т.д.

#### 4.4.5.

#### **Представление результатов активного потокового моделирования в виде конвейерно-временной диаграммы**

Каждый объект в классе проходит через 3 фазы: ожидает во входной очереди, реализует жизненный цикл (по одному из альтернативных вариантов), ожидает в выходной очереди. Далее объект в данном классе уничтожается, но передает в класс-потомок атрибут-идентификатор, а возможно, и некоторые другие наследуемые атрибуты. Цепочку объектов из разных классов, имеющих общий идентификатор (другие атрибуты), будем называть метаобъектом. Так, заказы в излагаемом здесь примере (см. рис. 4.13) являются метаобъектами.

Метаобъект проходит через систему по определенному маршруту, характеризуемому фазами пребывания объектов в классах и межклассовыми переходами. Метаобъекты запускаются в систему последовательно, один за другим, в так называемом конвейерном режиме. Одновременно может проходить, каждый свой маршрут, множество метаобъектов.

Ранее было показано (см. рис. 4.14), что объект в определенные моменты может воздействовать на характеристики системы, в частности потреблять ресурсы. Воздействие на характеристики системы осуществляется и множеством метаобъектов, работающих в параллельно-конвейерном режиме. Зависимость от времени изменений системных характеристик вследствие таких воздействий описывается конвейерно-временными диаграммами (КВД), формируемыми в результате моделирования.

Обобщим на случай активного потокового моделирования КВД, рассматривавшуюся нами ранее для пассивного моделиро-

вания (см. рис. 4.10). Обобщенная КВД состоит из графика в системе координат «время – характеристика системы» и набора так называемых метаобъектных шкал, размещенных под графиком параллельно оси времени. Каждая такая шкала соответствует маршруту одного метаобъекта. На ней кружками показываются граничные точки – моменты изменения фазы метаобъекта внутри классов и его перехода из класса в класс. Кружки тех граничных точек, в которых происходит изменение исследуемой характеристики системы, зачерняются.

Начальная и конечная точки шкалы обозначаются соответственно Н и К. Точки перехода метаобъектов из входной очереди на диаграмму жизненного цикла обозначаются символом начальной вершины этой диаграммы, а точки перехода из диаграммы жизненного цикла в выходную очередь – символом конечной вершины. Точки перехода метаобъекта из класса в класс обозначаются так же, как и на ЭПД, т.е. через  $t_i$ . Слева шкала помечается идентификатором соответствующего метаобъекта.

В примере на рис. 4.15 в качестве характеристики системы взяты финансовые затраты на обслуживание заказов. Принято, что финансы расходуются в точках перехода заказов (метаобъектов) из класса в класс – зачернены кружки  $t_i$ . Шкала с идентификатором  $a_{1,1} = X_1$  соответствует маршруту заказа, обеспеченного товарным ресурсом  $R_1$  (см. рис. 4.13). Заказ  $X_1$  ожидает во входной очереди класса  $\Pi_1$ , далее проходит жизненный цикл  $S_{1,1} - S_{1,5}$ , ожидает в выходной очереди  $\Pi_1$ , после чего переходом  $t_1$  передается в класс  $\Pi_2$  и т.д. Шкала с идентификатором  $a_{1,1} = X_2$  отображает маршрут отклоненного заказа (в классе  $\Pi_1$ ). Шкала с  $a_{1,1} = X_3$  соответствует маршруту необеспеченного заказа, который последовательно происходит через классы  $\Pi_1, \Pi_2, \Pi_4, \Pi_3, \Pi_5$ . Шкала с  $a_{1,1} = X_4$  повторяет шкалу для  $X_1$ , т.е. заказ  $X_4$  также обеспечен товарным ресурсом.

Заказы  $X_1, \dots, X_4$  запускаются в систему с некоторой задержкой один после другого, поэтому начальные точки соответствующих метаобъектных шкал последовательно сдвинуты вправо по оси времени. Из взаимного расположения метаобъектных шкал наглядно видно, какие заказы обслуживаются параллельно.

Зачерненные граничные точки метаобъектных шкал проецируются на временную ось графика  $Z = f(\tau)$  в виде моментов  $\tau_1, \dots, \tau_n$ . Значения ординат графика (финансовые затраты) для моментов  $\tau_i, i = 1, \dots, n$ , определяются в ходе имитационного моделирования ЭПД как значения соответствующих атрибутов.

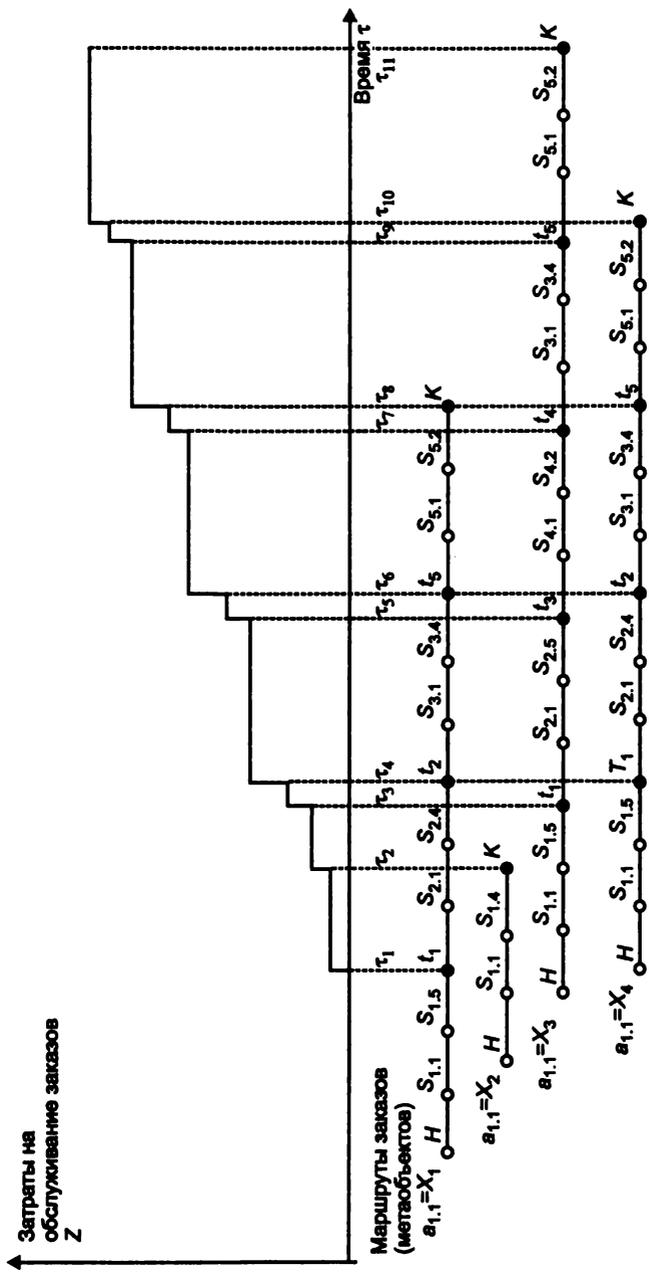


Рис. 4.15. Конвейерно-временная диаграмма для ЭПД (см. рис. 4.13)

Отметим, что в момент  $\tau_i$  значение ординаты графика может формироваться как результат воздействия нескольких параллельно обслуживаемых метаобъектов. В этом случае изменения характеристики системы, инициируемые разными метаобъектами, суммируются (в примере на рис. 4.15 – для моментов  $\tau_4, \tau_6, \tau_8$ ). На интервалах между моментами, нанесенными на ось времени, затраты на обслуживание заказов не изменяются – график конвейерно-временной диаграммы (см. рис. 4.15) представляет ступенчато-возрастающую функцию.

### Контрольные вопросы и упражнения

1. Как применительно к организационной системе интерпретируются понятия «дискретный поток» и «накопитель элементов потока»? В чем отличие пассивного и активного накопителей и соответственно пассивного и активного потокового моделирования? Что такое потоковый сценарий и на базе каких языковых средств составляется его описание? На какой математический аппарат опирается пассивное и активное потоковое моделирование? Приведите примеры объектов потокового моделирования – многоагентных иерархических организационных систем (МИОС). Как вы понимаете свободный и фиксированный порядок функционирования (следования операций) на различных уровнях МИОС? Опишите структуру базовой пассивной потоковой модели – триадной схемы и механизм взаимодействия ее частей. Охарактеризуйте базовую активную потоковую модель – эволюционно-потоковую диаграмму. Укажите задачи, которые решаются в рамках потокового моделирования организационных систем.
2. Приведите примеры графического описания потокового сценария МИОС и дополняющих его логических формул преобразования потоков. Как проявляется преемственность способа описания потокового сценария относительно нотаций стандарта SADT (технология структурного анализа и проектирования)?
3. Проиллюстрируйте на учебном примере методику пассивного потокового моделирования организационной системы, состоящую из следующих процедурных этапов:
  - построение на основе потокового сценария субмодели движения потоков (потоковой сети);
  - построение субмодели управления потоками (графов управления);
  - организация взаимодействия между потоковой сетью и графами управления при штатном и нештатном режимах работы системы;

- аналитическое и/или имитационное исследование потоковой модели с представлением результатов в виде конвейерно-временных диаграмм.
4. Изложите ключевую идею, положенную в основу методики активного потокового моделирования организационной системы. Для этого покажите, как потоковый сценарий процесса преобразовывается в эволюционно-потоковую диаграмму (ЭПД) и как проводится анализ ЭПД.

Для решения первой задачи:

- каждой операции сценария сопоставьте мобильные объекты (элементы потоков), обрабатываемые операцией, и укажите ресурсы, обеспечивающие ее жизнедеятельность;
  - введите атрибуты объектов и ресурсов;
  - в терминах атрибутов опишите внутриоперационные преобразования и жизненный цикл объектов;
  - задайте межоперационные преобразования атрибутов.
5. В рамках анализа ЭПД изложите методы построения конвейерно-временных диаграмм, оценки интегральных показателей динамики потоков, оценки достижимости целей по результатам имитационного моделирования.

# ПРИЛОЖЕНИЕ

## О проблемах логического моделирования и путях их решения

С 1996 г. при Институте проблем управления им В.А. Трапезникова РАН регулярно проводится общемосковский семинар «Логическое моделирование» ([http://www.ipu.rssi.ru/semin/so\\_yudit.htm](http://www.ipu.rssi.ru/semin/so_yudit.htm); E-mail: [vladislavlev@ Rambler.ru](mailto:vladislavlev@ Rambler.ru)), на котором заслушиваются доклады по различным аспектам системного анализа применительно к экономическим, техническим, социальным, административным и иным системам. Семинар посещают сотрудники ИПУ РАН и других научно-исследовательских организаций, преподаватели, аспиранты, студенты высших учебных заведений, работники компьютерных фирм и производственных предприятий.

Важное место в работе семинара занимает *поведенческое моделирование* как инструмент инжиниринга и реинжиниринга сложных систем, включая формализованное описание знаний о функционировании (поведении) системы, проведение имитационных экспериментов над поведенческими моделями, принятие по результатам экспериментов проектных решений. Поведенческие модели помогают также находить общий язык различным участникам процесса создания системы – специалистам в проблемной области (экспертам), системным аналитикам, программистам.

Ниже излагаются мнения нескольких ведущих участников семинара о проблемах логического моделирования и путях их решений.

### **Логическое и физическое моделирование** *Ю.Л. Барон, канд. техн. наук*

Говоря о логическом моделировании, следует иметь в виду не только начальную фазу (прототипирование, концептуальное моделирование и т.д.) построения моделей сложных систем, но и – более широко – одну из двойственных, взаимно дополняющих («логической» и «физической») составляющих процесса моделирования как такового. Необходимо сразу отметить, что разделение «логического» и «физического» моделирования носит чисто функциональный характер; границы этих составляющих не являются абсолютными. Данное разделение сугубо «рабочее» и используется на этапе проектирования и реализации конкретных моделей сложных систем. И тем не менее оно достаточно универсально и обоснованно. Рассмотрим ряд типовых относительных признаков, дискриминирующих «логическую» и «физическую» модельные составляющие.

#### «Логическая» составляющая

Явное представление системообразующих отношений, структурных параметров и правил взаимодействия процессов

Большое количество степеней свободы модельного описания

Высокая степень абстракции и универсальности описания

Относительная простота реконфигурации («подвижность») модельного описания

Приоритет режима анализа и интерпретации

Порождение высокоуровневого результирующего описания

Неоднозначность практической интерпретации результатов

#### «Физическая» составляющая

«Защитость» соответствующих параметров в теле модели

Ограниченное число степеней свободы (настраиваемых параметров)

Предметная ориентированность и конкретность

Относительная сложность и трудоемкость реконфигурации и перенастройки

Приоритет режима автоматического выполнения

Низкоуровневый протокол моделирования

Наличие однозначно вытекающих из результатов моделирования «физических» эффектов

Характерными при всей относительности данного разделения являются крайние позиции разработчиков моделей с преобладанием одной из указанных составляющих. С «физической» точки зрения процедурное содержание моделей есть главное, говорящее само за себя, и логическая надстройка нужна лишь для того, чтобы расширить область применения модели, сделать ее более гибкой, облегчить интерпретацию результатов и т.д.

С «логической» точки зрения не так уж важно, какое физическое (процедурное) содержание будет «подгружаться», наполнять выстроенную логическую структуру. Главный результат заключается в тех высокоуровневых и универсальных «объясняющих» оценках и высказываниях, которые удается сопоставить обобщаемому классу объектов и процессов.

Главная методическая (и практическая) цель выделения логической и физической модельных составляющих – подчеркивание возможности и необходимости независимого, но взаимосогласованного совершенствования арсенала соответствующих средств моделирования, использование двойственной (а на самом деле триадной – с учетом двусторонней связи между составляющими) схемы построения моделей. Чем более последовательно будет проводиться принцип взаимной дополнительности («объединяющего разделения») логической и физической модельных составляющих, тем более четкими смогут стать требования к каждой из этих составляющих (оцениваемой по своим критериям) и более мощными – результирующие объединенные модели. В качестве центральной может при этом рассматриваться проблема построения моделей поведения сложных систем (в частности, бизнес-систем), в качестве базового подхода – иерархическое конструирование системообразующих отношений: выдвигание нового уровня (мета-, субуровня) описания, уста-

новление двусторонних связей («отношений переноса») модельных описаний между новым и существующим уровнем; интеграция результатов в рамках целостной модельной конструкции.

Примерами рассматривавшихся с данных позиций системообразующих отношений могут, в частности, служить следующие парные (а на самом деле, как отмечалось, триадные) отношения: массовая – сверхмассовая задача; операционное (рабочее) – целевое (сценарное) пространство; модель – метамодель (в том числе на языке UML); исполнительская – управляющая структура (в том числе связанные отношением стимулирования); процесс развития – процесс мониторинга (в том числе социально-экономических систем).

С учетом накопленного опыта можно говорить о формировании определенной методологии логического моделирования в рамках упомянутой выше центральной проблемы. Для получения более интегрированного и практически значимого представления данной методологии необходимо системное (с единых позиций) рассмотрение следующих вопросов:

- выбор типовых практических задач;
- построение стандартизированной библиотеки базовых модельных пространств и типов отношений;
- поддержка отдельных этапов конструирования модели;
- поддержка сквозного процесса конструирования модели (с обеспечением функций «путеводителя» по «растущей» конструкции, ее анализа и отладки);

• экспериментальная проверка методологии на практических задачах.

В любом случае можно констатировать, что за время существования семинара его тематика, с одной стороны, вместила в себя ряд проблем, исследований, разработок из разных и на первый взгляд далеких научных и практических дисциплин, с другой – сохранила нацеленность на рассмотрение всего представленного многообразия задач и методов их решения с позиций именно логического моделирования. Правомерность такого подхода выдержала проверку временем. Остается пожелать не только развития, но и «замыкания», фиксации уже накопленного потенциала в виде более или менее стройной методологии логического моделирования поведения сложных систем.

### **Применение математических методов**

*Д.А. Новиков, д-р техн. наук, профессор*

Одной из проблем, которой в выступлениях на семинаре неизменно уделяется значительное внимание, является управление организационными системами, и в первую очередь использование математических моделей (использующих аппарат теории игр и теории оптимизации) при разработке процедур принятия управленческих решений.

На основании анализа докладов по проблемам управления организационными системами можно сделать вывод о наличии на настоящий момент двух тенденций. С одной стороны, при построении моделей организационных систем математиками, использующими привычный для них аппарат, получающиеся модели оказываются достаточно «узкими» и, как правило, без существенной адаптации не могут быть использованы

управленцами-практиками. Объяснением этому служит, очевидно, сложность моделируемого объекта. С другой стороны, при решении практических задач специалистами в соответствующих предметных областях зачастую не принимаются во внимание такие многие привычные для любого математика вопросы, как, например, корректность задачи – единственность решения, его устойчивость и т.д. Кроме того, большинство существующих сегодня инструментальных и программных средств автоматизации систем организационного управления основаны на имитационных моделях и, давая возможность как максимум проведения сценарного анализа, не позволяют решать прямые и обратные задачи управления: искать оптимальное управление или управление, обеспечивающее достижение поставленной цели с минимальными «затратами».

Возможным выходом, который позволил бы сгладить обе отмеченные тенденции, является более тесное сотрудничество специалистов-математиков и специалистов в предметных областях, в первую очередь на этапе постановки задачи и формировании требований к будущей модели.

### **Роль человеческого фактора**

*Н.А. Абрамова, д-р техн. наук*

В процессе моделирования человеком (экспертом, аналитиком и т.д.) выполняются такие формы деятельности, как формализация знаний о проблемных областях, когнитивный (познавательный) анализ ситуаций, идентификация результатов моделирования. В категорию приложений, для которых характерно участие человека, попадают методы и технологии искусственного интеллекта, поддержки принятия решений, экспертные технологии и др.

Адекватность предлагаемых наукой подходов к решению «человеческих» задач в моделировании зависит от научных представлений о человеке как субъекте познавательной деятельности, принятых в науках о человеке, в первую очередь в психологии. В связи с этим необходимы взаимообогащающие плодотворные контакты между специалистами по моделированию и психологами. Для психологов могут быть полезны:

- опыт работы с людьми, решающими практические интеллектуальные задачи с применением научных знаний и компьютерных средств;
- теоретические обобщения такого опыта и теоретические подходы к работе с экспертами;
- идентификация как составная часть научно-прикладной деятельности, которая является средством обеспечения адекватности теоретических представлений об объекте приложения знаний реальным свойствам объекта;
- исследования по психологической корректности формальных методов получения знаний от человека и по проблеме рисков, обусловленных влиянием познавательных средств человека в различных технологиях интеллектуальной деятельности;
- взаимопонимание в междисциплинарных коллективах, риск при использовании компьютерных средств поддержки интеллектуальной деятельности.

## ЛИТЕРАТУРА

---

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. – 2-е изд.: Пер. с англ. – М.: Бином; СПб.: Невский диалект, 1998.
2. Губко М.В., Новиков Д.А. Теория игр в управлении организационными системами. – М.: СИНТЕГ, 2002.
3. Заде Л. Понятие лингвистической переменной и его применение для принятия приближенных решений. – М.: Мир, 1976.
4. Когнитивный анализ и управление развитием ситуаций (CASC' 2001)// Материалы 1-й Междунар. конф: В 3 т. / Сост. В.И. Максимов. – М.: ИПУ РАН, 2001.
5. Когнитивный анализ и управление развитием ситуаций (CASC' 2002)// Материалы 2-й Междунар. конф: В 2 т. / Сост. В.И. Максимов. – М.: ИПУ РАН, 2002.
6. Котов В.Е. Сети Петри. – М.: Наука, 1984.
7. Ларичев О.И. Теория и методы принятия решений, а также хроника событий в Волшебных Странах. – М.: Логос, 2000.
8. Ларичев О.И., Мошкович Е.М. Качественные методы принятия решений. – М.: Наука, 1996.
9. Марка Д., Мак-Гоуэн К. Методология структурного анализа и проектирования. – М.: МетаТехнология, 1993.
10. Питерсон Дж. Теория сетей Петри и моделирование систем: Пер. с англ. – М.: Мир, 1984.
11. Робертс Ф.С. Дискретные математические модели с применением к социальным, биологическим и экономическим задачам. – М.: Наука, 1986.
12. Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и связь, 1993.
13. Таль А.А., Юдицкий С.А. Иерархия и параллелизм в сетях Петри // Автоматика и телемеханика. – 1982. – № 7, 9.

14. *Трахтенгерц Э.А.* Субъективность в компьютерной поддержке управленческих решений. – М.: СИНТЕГ, 2001.
15. *Юдицкий С.А.* Сценарно-целевой подход к системному анализу // Автоматика и телемеханика. – 2001. – № 4.
16. *Юдицкий С.А.* Сценарный подход к моделированию поведения бизнес-систем. – М.: СИНТЕГ, 2001.
17. *Юдицкий С.А., Владиславлев П.Н.* Предпроектное моделирование функционирования организационных систем. – М.: Научтехлитиздат, 2004.
18. *Юдицкий С.А., Радченко Е.Г.* Моделирование движения потоков при предпроектном анализе сложных систем // Приборы и системы. Управление, контроль, диагностика. – 2003. – № 9.
19. *Юдицкий С.А., Радченко Е.Г.* Построение потоковых моделей многоагентных иерархических систем // Приборы и системы. Управление, контроль, диагностика. – 2004. – № 2.
20. *Юдицкий С.А., Магергут В.З.* Логическое управление дискретными процессами. – М.: Машиностроение, 1987.

Учебное издание

**Юдицкий Семен Абрамович  
Владиславлев Павел Николаевич**

**ОСНОВЫ ПРЕДПРОЕКТНОГО АНАЛИЗА  
ОРГАНИЗАЦИОННЫХ СИСТЕМ**

Заведующая редакцией *Л.А. Табакова*  
Редактор *А.В. Тихонова*  
Младший редактор *Н.А. Федорова*  
Художественный редактор *Ю.И. Артюхов*  
Технический редактор *В.Ю. Фотиева*  
Корректоры *Т.М. Колпакова, Г.Д. Кузнецова*  
Компьютерная верстка *Е.Ф. Тимохиной,  
О.В. Фортунатовой*  
Оформление художника *Н.М. Биксентеева*

ИБ № 4827

Подписано в печать 03.03.2005  
Формат 60x88<sup>1</sup>/<sub>16</sub>. Гарнитура «Таймс». Печать офсетная  
Усл. п.л. 8,82. Уч.-изд. л. 7,87. Тираж 2000 экз.  
Заказ 684. «С» 069

Издательство «Финансы и статистика»  
101000, Москва, ул. Покровка, 7  
Телефон (095) 925-35-02. Факс (095) 925-09-57  
E-mail: mail@finstat.ru <http://www.finstat.ru>

ГП Псковской области «Великолукская городская типография»  
Комитета по средствам массовой информации,  
182100, Великие Луки, ул. Полиграфистов, 78/12  
Тел./факс: (811-53) 3-62-95  
E-mail: VTL@MART.RU