

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное автономное
образовательное учреждение высшего образования
«Пермский национальный исследовательский
политехнический университет»

Л.А. Мыльников, З. Аврамович, А.В. Мыльникова

**ИССЛЕДОВАНИЕ
ИНФОРМАЦИОННЫХ ПРОЦЕССОВ
ОРГАНИЗАЦИОННЫХ СИСТЕМ**

Монография

Издательство
Пермского национального исследовательского
политехнического университета
2023

УДК 005.7,65

ББК 16.2

М94

Рецензенты:

д-р техн. наук, профессор, *А.В. Иващенко*
(ООО «Открытый код», г. Самара);

д-р филос. наук, профессор, *С.В. Комаров*
(Пермский государственный национальный
исследовательский университет)

Мыльников, Л.А.

М94 Исследование информационных процессов в организационных системах : монография / Л.А. Мыльников, З. Аврамович, А.В. Мыльникова; ФГАОУ ВО «Пермский национальный исследовательский политехнический университет». – Пермь : Изд-во Перм. нац. исслед. политехн. ун-та, 2023. – 181 с.

ISBN 978-5-398-03074-7

В монографии рассмотрены концепции и подходы к проектированию, реконструкции и исследованию информационной инфраструктуры, представлен метод структурно-функционального моделирования и способ исследования получаемых моделей с выделенным субъектом управления, оценки информационных инфраструктур, определения целевых показателей организационных систем и контроля их достижения, элементы целеполагания, бенчмаркинга. В книге предложена новая нотация структурно-функционального моделирования (графическое и символьное представление), позволяющая исследовать с помощью методов моделирования эффективность информационной инфраструктуры (учитывать влияние субъектов управления на эффективность функционирования информационной инфраструктуры и цели организационной системы). Рассмотрены способы исследования эффективности реализации информационных процессов, а также способы генерации случайных чисел, подчиняющихся различным закономерностям, использование которых позволяет проводить исследование информационных инфраструктур с помощью методов имитационного моделирования. Рассмотрен способ реконструкции информационных инфраструктур из текстовых документов / сопроводительной документации с использованием скелетных структур текстов.

Издание подготовлено в результате реализации научного проекта № С-26/692 при финансовой поддержке Правительства Пермского края.

УДК 005.7,65

ББК 16.2

ISBN 978-5-398-03074-7

© ПНИПУ, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
Некоторые определения.....	9
Краткая история развития методов повышения эффективности функционирования организационных систем и исследования информационной инфраструктуры	11
Глава 1. ПЛАНИРОВАНИЕ И КОНТРОЛЬ В ОРГАНИЗАЦИОННЫХ СИСТЕМАХ (<i>Л.А. Мыльников</i>).....	15
1.1. Подходы к научно-техническому прогнозированию, прогнозированию трендов и значений параметров, выбору целей	15
1.2. Индикаторы и показатели достижения целей (KPI)	27
1.3. Оценка отклонений.....	34
1.4. Основы управления процессами и проектами.....	44
1.5. Механизмы повышения эффективности.....	46
Глава 2. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ (<i>Л.А. Мыльников</i>).....	51
2.1. Взаимосвязь эффективности и организации процессов	53
2.2. Моделирование информационных процессов с выделенным субъектом управления	58
2.3. Сравнение и выбор способов реализации информационных процессов	75
Глава 3. ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ ДЛЯ ИССЛЕДОВАНИЯ ПРОЦЕССОВ МЕТОДАМИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ (<i>З. Аврамович</i>).....	97
3.1. Способы получения случайных чисел	99
3.2. Равномерно распределенные случайные числа.....	103
3.2. Другие виды случайных распределений.....	117
3.4. Проверка принадлежности генератора случайных чисел и результатов моделирования чисел к заданному распределению	133

Глава 4. РАЗБОР ТЕКСТОВЫХ ИСТОЧНИКОВ ДЛЯ РЕКОНСТРУКЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ (<i>А.В. Мильникова</i>)	144
4.1. Взаимосвязь смысла и структуры текстов	144
4.2. Формализация структуры текстов на основе скелетных структур	152
4.3. Некоторые аспекты реконструкции информационных процессов из текстов	163
ЗАКЛЮЧЕНИЕ	172
СПИСОК ЛИТЕРАТУРЫ	173

ВВЕДЕНИЕ

Пока мы не вникнем мыслью в то, что есть, мы никогда не сможем принадлежать тому, что будет.

Мартин Хайдеггер

Процессы управления и поддержки принятия решений в условиях повышения качества используемых информационных систем приобретают элементы конкуренции, заключающиеся в качестве принимаемых решений. Таким образом, искусство управления становится искусством применения существующих инструментов в условиях повышающегося объема информации и меняющихся принципов взаимодействия между средствами производства и сотрудниками предприятий (табл. 1).

Таблица 1

Концепция развития индустрий

Конец XVIII века	Начало XX века	Конец XX – начало XXI века	2010 «+» годы	2030 «+»
Индустрия 1.0	Индустрия 2.0	Индустрия 3.0	Индустрия 4.0/ Advanced Manufacture	Индустрия 5.0
Использование силы воды и пара, других источников энергии	Массовое производство	Автоматизация и роботизация	Цифровизация процессов, IoT	Коворкинг с интеллектуальными системами и роботами (социальное взаимодействие)

Очевидно, для этого необходимо понимание процессов, протекающих в исследуемой системе, её структурное устройство и принципы взаимодействия между элементами. Именно изменение

этих принципов способно повысить эффективность управления и принимаемых решений в рамках сложных объектов управления.

Современные производственные системы необходимо рассматривать как гетерогенные системы, претерпевающие изменения во времени. Сложность таких систем обусловлена числом компонент (подсистем) и их взаимосвязей.

В связи с отсутствием инструментария управления разнородными системами управление сложными гетерогенными системами осуществляется путем реализации программ и проектов развития в широком смысле.

Традиционное понимание процесса принятия решений включает: определение целей, сбор и анализ информации, формирование множества альтернатив, разработку системы критериев и показателей для оценки альтернатив и мониторинга процесса достижения цели. В таком контексте в настоящее время организации рассматривают своё текущее и будущее состояние (концепция VUCA [1]). Они должны максимально точно представлять границы для планирования и управления. Управление производственными системами, ориентированными на выпуск новой продукции, зависит от типа и числа проектов. В таких системах необходимым является определение частоты выпуска новых проектов и вывода старых, возможные отклонения в характеристиках новых проектов и параметрах рынка, при которых система остаётся экономически эффективной.

Параметры внешней среды и характеристики проектов являются сложно измеримыми и содержат ошибку. С. Сэвидж считает, что такие системы стоит рассматривать как стохастические [2].

Системы, функционирующие в описанных условиях, относятся к классу «мягких» систем. Для анализа «мягких» систем П. Чекладом [3] была предложена методология «мягкого» системного анализа. Эта методология является системно организованным процессом исследования плохо определенной системы, включающим в себя ряд последовательных этапов для принятия решения по управлению ситуацией.

Д. Марчем, М. Козном и Дж. Олсеном предложена модель garbage can [4] для принятия решений в условиях неопределенности.

Г. Саймон для управления в условиях неопределенности предложил теорию ограниченной рациональности, которая позволила объяснить разницу в принимаемых решениях крупными корпорациями, и рекомендации, которые дают теории рационального поведения [5].

Для управления «мягкими» системами широкое распространение получили гибкие методологии управления, такие как AGILE и SCRUM [6], KANBAN, Design Thinking [7], Lean Production, Six Sigma (шесть сигм).

Для описания информационной инфраструктуры используются методы структурно-функционального моделирования, а именно eEPC [8], BPMN, концепции WfMS [9], UML, OPM [10], SysML, ArchiMate [11].

Для исследования их функционирования используют сети Петри [12], процессные модели, методы имитационного моделирования (системная динамика Дж. Форрестера [13]); Вил ван дер Аалст предлагает использовать подход, получивший название, Process Mining [14].

Отдельно стоит отметить методы модификации и управления организационной структурой информационной инфраструктуры. Широко используемыми подходами являются подходы AS IS–TO BE, BPM, а также подходы, основанные на использовании удачных решений, – ITIL, COBIT

В общем виде рассматриваемая задача является NP-трудной. На практике рассматриваются отдельные случаи, связанные с управлением и принятием решений. За развитие этих методов в одной из прикладных областей Ж. Тироль получил в 2014 году Нобелевскую премию [15]. П.О. Скобелев для приближенного решения NP-полных задач разработал концепцию эмерджентного интеллекта [16].

Существующие в настоящее время организационные системы – это системы, которые смогли найти устойчивое состояние в сложившихся условиях. Изменение внешних и/или внутренних условий приводит к уходу из поля зрения одних систем и появлению новых. Системы, которые сохранили свою жизнедеятельность в но-

вых условиях, должны были перейти в новое устойчивое состояние. Так как внутренние и внешние возмущающие воздействия происходят постоянно, то существующие системы необходимо рассматривать как диссипативные системы, непрерывно переходящие из одного устойчивого состояния в другое (адаптивно-диссипативные системы). Механизмом перехода в новое устойчивое состояние является факт принятия управленческих решений в области применения новых методов, структурной организации протекающих процессов (схемы применения существующих и новых методов).

Таким образом, динамика состояний организационной системы будет описываться схемой, представленной на рис. 1. При этом функция поиска новых состояний связана с использованием информационной инфраструктуры.

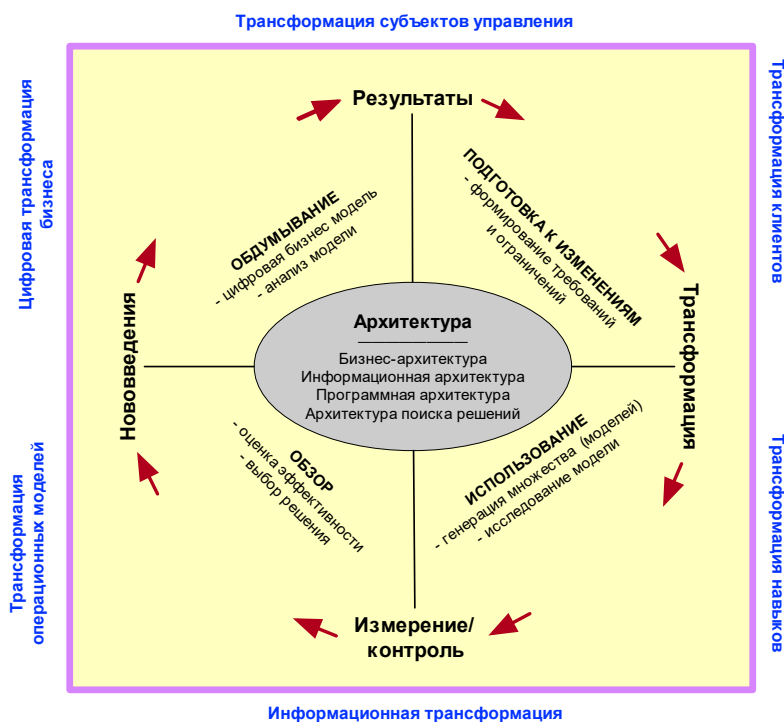


Рис. 1. Цикл непрерывной цифровой трансформации

Данная книга рассматривает подходы управления производственными системами, начиная с этапа определения цели, составления программ, контроля их выполнения, и рассматривает модели и методы, которые помогают произвести необходимые изменения в условиях ограниченных ресурсов.

Некоторые определения

Бенчмаркинг (нем. «контроллинг») – сопоставительный анализ на основе эталонных показателей процессов, протекающих в рассматриваемой системе для улучшения её работы (определение, понимание и адаптация процессов).

Гетерогенная система – неоднородная система, состоящая из однородных частей (подсистем), различающихся структурой/ организацией / описанием / используемыми моделями, методами, принципами управления / протекающими в них процессами/способами представления и формализации / используемыми переменными и показателями.

Информационная инфраструктура – совокупность моделей, методов, алгоритмов обработки информации, аппаратного обеспечения и способ их использования (последовательность применения / структура).

Ключевые показатели эффективности (англ. KPI) – числовые показатели деятельности.

Лицо, принимающее решение (ЛПР, субъект управления), – индивидуум или группа лиц, которая не удовлетворена существующей ситуацией или развитием ситуации и имеет желание и возможность для инициации действий по изменению ситуации.

Неопределенность – неполнота или неточность информации об условиях реализации процесса или проекта.

Организационная система – совокупность элементов и процессов (гетерогенная система), ориентированных на достижение общей цели.

Повторяемые процессы – процессы, выполнение которых повторяется несколько раз и исполнение которых происходит в условиях уже сформированной организационной структуры. Таким об-

разом, генерируются только результаты, которые были заданы заранее, а последовательность выполнения задач может варьироваться только в пределах predetermined альтернатив.

Принятие решений организационной системой – процесс, предпринимаемый организационной системой для того, чтобы улучшить свое будущее состояние.

Проект¹ – это разовая инициатива, которая охватывает несколько подсистем организации, имеет ограниченную временную рамку, ориентирована на специальную задачу, является значимой и насущной и не поддается простому управлению в рамках существующей структуры организации, а требует вместо этого специальных организационных мер и инструментов для реализации. Признаки проекта заключаются в том, что они реализуются, как правило, один раз, реализуют комплекс мероприятий и затрагивают множество сфер системы, в которой реализуются, имеют лимитированные ресурсы (бюджет, материалы, персонал), имеют заранее заданную измеримую цель и строго заданный срок окончания.

Проект в узком смысле – набор мероприятий, направленных на создание конкретного изделия, ранее не выпускаемого в рассматриваемой производственной системе.

Проект в широком смысле – группа проектов в узком смысле, закрывающая определенную потребность.

Производственная система (ПрС) – система, осуществляющая все необходимые мероприятия по выпуску продукта, включая организацию производственного процесса, изготовление и сборку [17].

Производственно-экономическая система – система, включающая в себя производственную систему и самостоятельно осуществляющая сбытовую деятельность на открытом рынке [17].

Процесс – это набор видов деятельности или задач, которые необходимо выполнить частично или полностью в определенной последовательности, чтобы получить результат. Процессы могут требовать результатов выполнения других процессов.

¹ Проект происходит от латинского *proiectum*, что означает выдающийся / брошенный вперед = что-то, что было запланировано, поддается управлению и будет реализовано.

Разовые/единовременные процессы называют **проектами**.

Система – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство.

Краткая история развития методов повышения эффективности функционирования организационных систем и исследования информационной инфраструктуры

Эпоха возрождения – венецианская двойная бухгалтерия (Л. Пачоли).

1600–1800 годы – с появлением мануфактур и книгоиздания в отдельное направление выделилась экономика коммерции и торговли с уклоном на выявление факторов, приводящих к успеху с учетом производственных факторов (Жак Савари) [18].

В 1898 году была открыта первая высшая экономическая школа: сначала в Лейпциге, затем в Вене, Сан-Галлене и Ахене [19].

В 1900 году Анри Файоль впервые сформулировал функции управления (планирование, организация, распоряжение, координация, контроль), которые подробно описал в работе «Общее и промышленное производство», вышедшей в 1916 году. Развитием данного подхода считают циклы Деминга и Шухарта, которые были предложены в первой половине 1940-х годов [18] (рис. 2).

Алберт Калмес в 1906 году опубликовал книгу «Фабричное производство», в 1909 году книгу «Фабричная бухгалтерия», а в 1911 году – «Статистика в фабричном и товарном производстве», ставшие основой для развития методов управления основанных на данных (см. рис. 2).

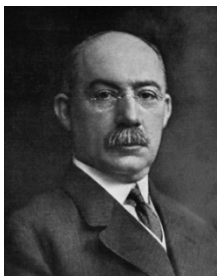
В начале XX века Генри Гант предложил метод наглядного упорядочения труда, получившего название диаграмма Ганта (см. рис. 2).

1919 год – экономическая школа с ориентацией на промышленное производство в г. Нюрнберге [19].

В 1930-е годы на основе трудов Ж. Фурье, Дж. фон Неймана и Л.В. Канторовича выделилась новая область знаний, получившая название «исследования операций».



Анри Файоль
(29.06.1841–19.11.1925),
Турция, Франция



Генри Гант
(20.05.1861–23.11.1919),
США



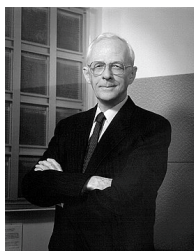
Альберт Калмес
(26.02.1881–22.09.1967),
Люксембург, Германия

Рис. 2. Пионеры информационного управления в промышленности

Середина 1950-х годов – создание Джейм Форрестером метода изучения сложных систем, получившего название «системная динамика» (рис. 3).

1950–1990 годы – появление методов планирования и управления производством и информационных систем на их основе: Just-in-Time, MRP, ERP, ERP II, CALS, OPT.

В 1962 году Карл Петри предложил способ описания протекания процессов, получивший название «сеть Петри» и используемый в настоящее время для моделирования дискретных динамических систем (см. рис. 3).



Джей Райт Форрестер
(14.07.1918–16.11.2016),
США



Карл Адам Петри
(12.07.1926–02.07.2010),
Германия

Рис. 3. Ученые, внесшие вклад в развитие методов исследования функционирования информационных процессов

1960–1980 годы – развитие методов многокритериального ранжирования (1968 год: Бернард Рой предлагает метод ELECTRE; 1979 год: Серафим Оприцович разрабатывает метод VIKOR; в 1982 году появляется первая реализация метода PROMETHEE, а в 1983 году – первый вариант реализации метода TOPSIS) (рис. 4).

1970–1980 годы – время появления классических методологий управления (модели водопада, спиральной модели, концепции design thinking).

В 1970-х годах Ричард Нолан сформулировал понятие зрелости системы в области обработки информации и выделил шесть стадий: 1) инициирование, 2) распространение, 3) контроль и управление, 4) интеграция, 5) ориентирование данных, 6) зрелость.

В 1976 году Томас Маккейб предложил для оценки сложности алгоритмов использовать цикломатическую сложность.

Вторая половина 1970-х годов – создание Питером Пин-Шан Ченом первой нотации графического описания системы хранения и передачи информации (появление ER модели), а также способа графического представления потока данных (DFD, data flow diagram).



Бернард Рой
(15.03.1934–28.10.2017),
Франция



Томас Саати
(18.07.1926–14.08.2017),
Ирак, США

Рис. 4. Ученые, внесшие существенный вклад в развитие методов выбора решений

Период конец 1970–1990-е годы: появление классических методов оценки отклонений, применяемых в бенчмаркинге, таких как DEA (Data Envelopment Analysis), метод оценки эффективности,

геометрические подходы оценки расстояний и отклонений, использование взвешенных сумм.

1980 год – время появления разработанного Томасом Саати метода анализа иерархий для учета предпочтения лиц, принимающих решения и его развитие в 2000-х годах и появление метода анализа сетей (см. рис. 4).

1980–2010 годы: время появления методологий структурно-функционального моделирования (IDEF, EPC и BPMN, предложенные Августом Шеером в рамках методологии ARIS; OPM, предложенной Дэйвом Дори, и ArchiMate, разрабатываемый The Open Group в рамках концепции TOGAF) (рис. 5).

В 1991 году Software Engineering Institute (SEI) американского университета Карнеги – Меллона предложил модель организационной зрелости (Capability Maturity Model).

1990–2010 годы: время появления гибких методологий управления AGILE и её развитие Джеффом Сазерлендом (методология SCRUM) (см. рис. 5).



Август-Вильгельм Шеер
(27.07.1941–н.в.),
Германия



Джефф Сазерленд
(20.06.1941–н.в.),
США



Вил ван дер Аалст
(29.01.1966–н.в.),
Нидерланды, Германия

Рис. 5. Ученые, внесшие существенный вклад в формализацию процессов управления и принятия решений

2000–2020 годы: создание и развитие профессором Вилом ван дер Аалстом нового направления исследования информационных процессов, получившего название Process Mining (см. рис. 5).

Глава 1. ПЛАНИРОВАНИЕ И КОНТРОЛЬ В ОРГАНИЗАЦИОННЫХ СИСТЕМАХ*

Нет ничего более бесполезного, чем эффективно делать то, что вообще делать не следует.

Питер Друкер

1.1. Подходы к научно-техническому прогнозированию, прогнозированию трендов и значений параметров, выбору целей

Правильная постановка цели напрямую влияет на успех или неудачу в достижении задуманного. Неправильно выбранная цель будет приводить к улучшению, а не к развитию.

Наблюдая за историей цивилизаций Европы и Азии, можно заметить, что во многом они не были первыми. Например, аборигены из Новой Гвинеи первыми стали обрабатывать землю, но потом, вероятно, их цивилизация развивалась не тем путем. Путем, который приводит к постоянному улучшению, но не развитию (их каменные топоры непрерывно совершенствуются уже несколько тысячелетий, а разнообразие языков достигает 900 штук и 19 языковых групп). Считается, что развитие проявляется не в улучшении, а в усложнении.

Бизнес-структуры и другие организации служат для достижения менее масштабных целей. Владелец организации определяет цели, а управленческая команда должна сделать все возможное для их достижения. Однако на практике выбор цели является непростой задачей, сложность решения которой связана с тем фактом, что среда, в которой функционирует система, меняется; в этой среде взаимодействует множество игроков, каждый из которых может по-

* Глава написана Л.А. Мыльниковым, Пермский национальный исследовательский политехнический университет.

разному представлять и учитывать то, как будет меняться ситуация; реализуется множество проектов с разными целями и временем реализации, присутствует информационный шум.

Таким образом, для формулирования цели **нужна гипотеза** о том, как может развиваться система (если мы сами планируем оказывать существенное влияние на окружение) или как будет развиваться внешний мир (если мы планируем связать свое развитие со внешними изменениями, которые мы ожидаем). Например, мы выдвигаем гипотезу, что кто-то в книге, которую мы читаем, совершил преступление, тогда целью может быть выяснение того, кто и как это сделал.

Постановка гипотез для производственных систем в ситуациях, когда мы хотим изменить наше окружение за счет выпуска новой продукции, связана с подходами к выработке инновационных решений и продуктов.

Для этого используется модель Клейтона Кристенсена, получившая название **модели «подрывных» инноваций**. Она объясняет, почему компании-лидеры терпят крах с приходом новых технологий. «Подрыв» происходит, когда покупатели начинают использовать инновационные продукты и услуги, даже если они несовершенны. Со временем покупателей становится больше, технология начинает активно развиваться, и в итоге новички захватывают рынки. Таким образом, достижение цели основано на новом качестве или свойстве продукта / услуги / ..., которые мы предлагаем.

Другой подход – это так называемое **дизайн-мышление**, которое предлагает решать бизнес-задачи, ориентируясь в первую очередь на интересы пользователя. Задача команды: определить проблему, понять, какая «боль» у потенциальных клиентов и как ее облегчить.

Дальше участники генерируют идеи, много идей, и – самое важное – на основе лучшей идеи создается прототип. При этом ценятся не только практические, но и эмоционально наполненные задумки, всячески поощряется командная работа и творческий подход, а логике авторы концепции предпочитают интуицию.

Методология Design Thinking возникла как средство решения проблем с точки зрения заказчика или пользователя для лучшего определения требований пользователя (рис. 6).

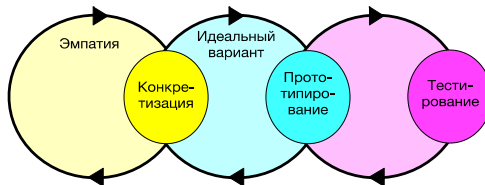


Рис. 6. Схема реализации дизайнерского мышления (design thinking) [17]

Популярный инструмент бизнес-стратегии разработал американский экономист Майкл Портер, который получил название **пяти сил Портера**.

Подход помогает понять, стоит ли вообще тратить силы на захват нового рынка и как сохранить позиции на уже существующем. Для этого Портер предлагает проанализировать пять факторов: угрозу появления новых игроков и продуктов-заменителей, уровень конкуренции, а также понять степень влияния поставщиков и потребителей на выбранный сегмент рынка.

Аналогично можно использовать методы **SWOT**, **PEST**, **SNW** анализа. SWOT – это сокращение от Strengths («сильные стороны»), Weaknesses («слабые стороны»), Opportunities («возможности»), Threats («угрозы»). При проведении анализа составляют таблицу подобную шаблону, приведенному в табл. 4.

Таблица 2

Таблица для проведения SWOT-анализа

Параметр	Положительное влияние	Отрицательное влияние
Внутренняя среда	Strengths (свойства проекта или системы, дающие преимущества)	Weaknesses (свойства, ослабляющие проект или систему)
Внешняя среда	Opportunities (внешние вероятные факторы, дающие дополнительные возможности по достижению цели)	Threats (внешние вероятные факторы, которые могут осложнить достижение цели)

Аналогичный SWOT-анализу подход получил название SNW-анализ (сокращение от Strength, Neutral, Weakness). Это анализ сильных, нейтральных и слабых сторон организации. В ситуации стратегического анализа в качестве нейтральной позиции, как правило, фиксируют среднерыночное состояние для рассматриваемой системы.

Еще одной разновидностью является PEST-анализ, в рамках которого выделяют политические (Political), экономические (Economic), социальные (Social) и технологические (Technological) аспекты внешней среды, которые влияют на бизнес компании.

Введение собственной стратегии поведения подразумевает не только формирование цели, не подразумевающей встраивание в стратегии других участников, но и отстаивание / борьбу за свою позицию.

Для предугадывания или проверки выработанных решений используют методы экспертного прогнозирования, имитационное моделирование для исследования систем (реакция на варианты воздействия, определение воздействия по желаемой реакции), статистические методы обработки данных (работа с временными рядами, выявление правил, трендов, значений показателей).

Среди методов экспертного оценивания наиболее известными являются методы Делфи и метод сценирования.

Метод Делфи был разработан в 1950–1960 годах в США корпорацией RAND, авторами считаются Олаф Хелмер, Норман Дэлки и Николас Решэр.

Суть этого метода в том, чтобы с помощью серии последовательных действий – опросов, интервью, «мозговых штурмов» – добиться максимального консенсуса при определении правильного решения. Анализ с помощью дельфийского метода проводится в несколько этапов, результаты обрабатываются статистическими методами (такими, как голосование большинством, метод Борда, метод Кондорсе, метод с двумя турами [17] и др.).

Другой подход – это **метод сценирования**. Его автором считают Германа Кана. Классическая методика сценирования заключа-

ется в выделении базовых количественных и качественных тенденций развития и отслеживании возможных изменений под их воздействием. Средние показатели тенденций образуют «базовый сценарий», то есть общую линию развития событий. Вариации в показателях и динамике тенденций образуют дополнительные или альтернативные сценарии. Особенность сценарного метода в том, что он позволяет определять направления и варианты развития событий, в том числе на среднесрочную и долгосрочную перспективу (при корректно указанных и отслеженных трендах), однако плохо предсказывает даты и точный формат событий [17].

Классическое сценирование опирается на получение «многоаспектного тренда» развития мира, включающего в себя некоторое количество независимых тенденций развития. На основе его анализа строится «прогноз без неожиданностей» (surprise free projection), или базовый сценарий развития – восемь канонических вариаций, образующих контекст базового сценария.

Альтернативное сценирование основано на неизменности прошлого и вариантности будущего. Сценарные ветви альтернативны. Сценарии не взаимодействуют между собой. Ключевые понятия, которыми оперируют, являются сценарная развилка, окно возможностей, сценарный выбор (акт управления) (рис. 7).

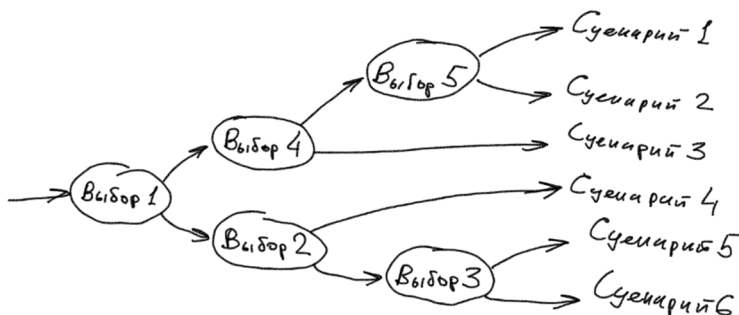


Рис. 7. Принцип альтернативного сценирования

Простейшее бизнес-сценирование основано на рассмотрении нескольких основных вариантов (табл. 3) в местах выбора.

Принцип формирования группы стратегий
при простейшем бизнес-сценировании

Параметр	Ничего не делать	Делать что-то
Условия обычные или неблагоприятны	Ничего не делаем, ничего хорошего и не получается	Делаем то, что умеем. Что-то получается, но ситуация все время ухудшается, хотя до катастрофы дело не доходит
Условия обычные или благоприятны	Ничего не делаем, но Господь милостив, и все получается прекрасно	Делаем все, что в наших силах, и выстраиваем новую систему внутри и снаружи

Продвинутое сценирование работает с тремя категориями управляющих воздействий: тренды; драйверы, или метатренды, учет упущенных сценарных ветвей; «дикие карты» – внешние по отношению к системе возмущения.

При континуальном сценировании число сценариев бесконечно, но любой представлен как линейная комбинация нескольких предельно возможных сценариев. Определено неизбежное будущее – общее ядро всех сценариев. Разрабатываются возможные версии развития такого ядра. Выбираем базовый сценарий, остальные сценарии рассматриваются как риски базового.

Сценирование осуществляется на основе рационального выбора. Рациональность связана как с аргументацией от определенных допущений к выводам, так и со способностью специалиста по планированию защищать или объяснять допущения и выводы.

Примеры формирования стратегий / целей (вариантов развития мира) такого рода можно найти в литературе, а именно в фантастике Роберта Хайнлайна, Станислава Лема, Герберта Уэллса, Алексея Толстого, Айзека Азимова, Аркадия и Бориса Стругацких, Герберта Франка, Александра Беляева, Джорджа Оруэлла, Артура Кларка и других, а также в научно-фантастических фильмах, таких как «Стар Трэк», «Звездные войны», «Тенет», «Интерстеллар» и т.д.

Простейший пример борьбы стратегий можно продемонстрировать примером из кинематографа СССР. В 80-е годы один фильм

показывали в разных кинотеатрах с разными концовками. Люди, которые посмотрели фильм, спорили о том, какой действительно конец имеет фильм. Победа не всегда оказывалась на одной стороне. «Предугадать развитие событий и извлечь из этого выгоду» может являться еще одним способом выбрать цель.

Для оценки значений параметров и скорости развития рассматриваемых сценариев необходимо использование методов имитационного моделирования. Построив модель, исследователь может: прогнозировать свойства и поведение объекта как внутри области, в которой построена модель, так и (при обоснованном применении) за ее пределами (прогнозирующая роль модели); управлять объектом, отбирая наилучшие воздействия путем испытания их на модели (управляющая роль); познать явление или объект, модель которого он построил (познавательная роль модели); получать навыки по управлению объектом путем использования модели как тренажера или игры (обучающая роль); улучшить объект, изменяя модель и испытывая ее (проектная роль).

Общеизвестно, что изучение аэродинамических свойств самолета производится, кроме всего прочего, в аэродинамической трубе, куда помещается сначала уменьшенная копия самолета, а на заключительном этапе исследований и сам самолет. При воздействии на объект воздушного потока проверяется, как на разных скоростях полета воздух обтекает самолет. Таким образом, устанавливают – оптимальна ли форма самолета, и надо ли ее дорабатывать. Другое применение аэродинамических труб – это продувка автомобилей. Продувка макетов кораблей также позволяет судить об их ветровых качествах, хотя скорость корабля намного меньше, чем у самолета или автомобиля, но и ветер на водной поверхности достигает большего значения. Моделирование используется и на макетах зданий, например, с целью проверки на ветроустойчивость. В конце 80-х годов было проведено моделирование ситуации, которая могла наступить вследствие массированного применения ядерного оружия, хотя моделирование проводилось независимо и советскими и американскими учеными, был получен одинаковый результат: ядерная зима и т.д.

Аналогично можно поступить и при производственных системах, заменяя материальные потоки потоками данных (см. системную динамику Дж. Форрестера [13]), и информационных системах.

Для оценки задач поддержки принятия решений в «мягкой» постановке в условиях динамики, задача определяется как задача оценки изменения значений параметров и/или факторов на основе субъективной модели (например, экспертных оценок значений факторов). Такая субъективная модель может быть формализована в виде ориентированного графа, который называют **когнитивной картой**.

Пример когнитивной карты приведен на рис. 8. На рисунке вершины графа – это факторы, а дуги – причинно-следственные связи между ними. Знак «плюс» у связей между вершинами-факторами означает, что увеличение значения фактора-причины приводит к увеличению значения фактора-следствия (прямая связь), знак «минус» означает обратную зависимость. Когнитивная карта отражает функциональную структуру анализируемой системы или процесса.

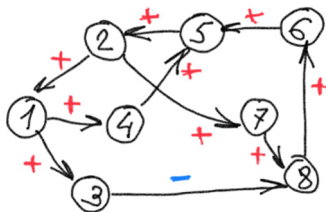


Рис. 8. Пример когнитивной карты (номераи обозначены узлы, которые соответствуют оцениваемым параметрам)

Процесс получения когнитивной карты является итерационным. Его этапы выполняются до тех пор, пока полученная структура не будет правдоподобно описывать динамику развития изучаемой системы или процесса [20]:

1. Генерация гипотетической когнитивной карты (выдвижение гипотезы о функциональной структуре когнитивной карты), которая включает:

– **создание абстрактной когнитивной карты**: определение множества параметров/факторов и причинно-следственных связей между ними;

– **параметризация когнитивной карты**: определение численных значений (коэффициентов) причинно-следственных связей (в результате выполнения этой операции абстрактная когнитивная карта превращается в модель конкретной динамической системы).

2. Верификация когнитивной карты (проверка соответствия гипотезы о функциональной структуре реальной системе или процессу) заключается в построении на полученной модели прогноза на основе ретроспективных данных и его сопоставления с имеющимися данными. Такая проверка позволяет выявить несогласованность между получаемыми данными и реальным развитием ситуации в прошлом. Верификация может быть прямой – проигрывание ситуации в будущее, и обратная – оценка развития ситуации при движении времени в прошлое.

3. Корректировка когнитивной карты (функциональной структуры) осуществляется по результатам верификации. Она может состоять в изменении: а) шкалы факторов, б) значений коэффициентов связей между узлами / факторами, в) добавлении или удалении новых узлов / факторов и/или связей между ними. Такие изменения могут оказать сильное влияние на получаемые результаты (так как когнитивная карта представляет собой ориентированный граф), что потребует новой верификации и, возможно, корректировки.

Итерационный процесс построения когнитивной карты повторяется до получения функциональной структуры, согласующейся с поведением изучаемой системы.

Если значения всех необходимых для анализа ситуации факторов нам известны, то для прогнозирования значений переменных могут использоваться методы регрессионного анализа, такие как метод наименьших квадратов, *lasso*, метод частичной регрессии наименьших квадратов (PLS), методы машинного обучения применительно к задачам регрессии (деревья решений, метод опорных векторов, *k*-ближайших соседей и т.д.).

Таким образом методы моделирования и прогнозирования позволяют оценить величины значений, которые мы могли бы достичь и тем самым на их основе сформулировать цели.

В институциональной экономике на операционном уровне получил распространение подход, основанный на использовании производственных функций. Организация рассматривается как «черный ящик», которому соответствует производственная функция [21].

Теория производственных функций посвящена изучению и функциональному описанию взаимодействия между производственными системами и проектами с учётом различных факторов и описывает работу ПрС в виде математической модели – формулы зависимости выпуска продукции (объёма или дохода) от вектора расходуемых или используемых в производстве ресурсов (затрат на их покупку) [22, 23]. Среди получивших наибольшее распространение в рамках данной теории следует отметить функцию с фиксированными пропорциями факторов (производственная функция (ПФ) Леонтьева), ПФ Кобба – Дугласа, линейную ПФ, ПФ Аллена, ПФ с постоянной эластичностью, ПФ с линейной эластичностью замены факторов, ПФ Солоу или Хилхорста, ограниченную функцию, много режимную функцию, ПФ линейного программирования [24–26].

Такой способ оценки поведения фирм принадлежит американскому экономисту Дж.Б. Кларку. В этом случае ПФ будет формировать обобщенную критериальную функцию, на основе которой могут быть определены достижимые значения показателей [27].

Из литературы нам известно несколько рекомендаций к постановке целей, самая известная из которых – это постановка цели по SMART (постановка умных целей) [28]. В рамках этого подхода предполагается, что цель должна удовлетворять ряду требований (табл. 4).

Другой подход, подход предложенный П. Друкером, – это опора на такие базовые измеримые показатели, как время, стоимость и качество. Эти показатели должны стать базой для формулировки цели или стратегии (если рассматривать её как множество разнесенных во времени целей). Тогда сформировав ожидания к ним,

можно будет определить направление развития / движения, которые и станут целью или, как принято говорить в рамках этого подхода, стратегией.

Таблица 4

Постановка цели по SMART

	Параметр	Характеристика
S	Specific (конкретный)	Объясняется, что именно необходимо достигнуть
M	Measurable (измеримый)	Объясняется, в чем будет измеряться результат
A	Attainable, Achievable (достижимый)	Объясняется, за счёт чего планируется достигнуть цели. И возможно ли её достигнуть вообще?
R	Relevant (актуальный)	Определение истинности цели. Действительно ли выполнение данной задачи позволит достичь желаемой цели?
T	Time-bound (ограниченный во времени)	Определение временного промежутка, по наступлению/окончанию которого должна быть достигнута цель

Стратегия «голубых океанов» говорит нам, что для постановки целей необходимо сформулировать отличия. Для этого предлагается ответить на четыре базовых вопроса: 1) что убрать? 2) что уменьшить? 3) что создать? 5) что усилить?

Некоторые авторы предлагают выбирать цель, исходя из видов эффектов, которые необходимо получить (табл. 5).

Таблица 5

Виды эффектов [18]

Экономический эффект	Научно-технический эффект	Социальный эффект	Экологический эффект
Прибыль от лицензионной деятельности	Количество зарегистрированных результатов интеллектуальной деятельности	Прирост доходов работников	Снижение выбросов в атмосферу, почву, воду вредных веществ
Прибыль от внедрения патентов, изобретений, ноу-хау	Увеличение удельного веса новых технологий	Повышение уровня безопасности труда	

Окончание табл. 5

Экономический эффект	Научно-технический эффект	Социальный эффект	Экологический эффект
Прирост объема продаж	Увеличение удельного веса прогрессивных технологических процессов	Повышение степени удовлетворения социальных и духовных потребностей	Снижение отходов производства
Улучшение использования производственной мощности	Повышение коэффициента автоматизации производства	Увеличение числа рабочих мест	Повышение эргономичности производства и продукции
Сокращение срока окупаемости инвестиций	Повышение организационного уровня производства и труда	Повышение квалификации работников	Повышение экологичности производства и продукции
Сокращение сроков капитального строительства	Повышение конкурентоспособности организации и ее товаров на рынке	Улучшение условий труда и отдыха работников	Снижение штрафов за нарушение экологического законодательства и других нормативных документов
Улучшение использования ресурсов: – рост производительности труда; – повышение фондоотдачи; – ускорение оборачиваемости средств			

Для облегчения работы управленческой команды используют различные подходы и методы управления [29]. При этом прозрачность является одним из важных показателей, указывающим на уровень достижения поставленной цели и фиксирующим отклонения от выполнения поставленных целей.

Поставленные цели и решаемые для их достижения задачи не статичны, а изменчивы во времени. Поэтому, для того чтобы оставаться успешной, организация должна меняться сама. Она должна расти, становиться более эффективной (прибыльной), сокращать затраты, выпускать новые продукты, предлагать новые услуги и т.д., из чего следует, что организация должна развиваться по инновационному пути [30]. Ей следует находить новые и лучшие способы для достижения своих целей. Необходимо использовать показатели эффективности управления процессами и проектами, которые, в свою очередь, требуют соответствующих рычагов управления.

Организации не только обеспечивают функционирование существующих процессов, но и выполняют проекты с уникальными результатами, которые открывают новые и усовершенствованные виды деятельности. В связи с тем, что под инновацией подразумевается создание чего-то нового, управление изменениями представляется подходящим инструментом для управления инновационными видами деятельности [30].

1.2. Индикаторы и показатели достижения целей (KPI)

Объект управления находится в пространстве состояний. Координатами n -мерного пространства состояний выступают параметры объекта управления, которые являются важными для достижения управленческих целей. Значения параметров показывают, насколько эффективно состояние объекта управления для достижения выбранной цели.

Цели, которые необходимо достичь, называют целевым вектором z_p , а текущее состояние объекта управления описывается вектором z_a . Введем оценку $d(z_a, z_p)$, которая будет расстоянием между целевым и текущим состояниями объекта управления в рассматриваемом пространстве состояний. Состояние объекта управления является ключевым индикатором эффективности управления. Если $z_p = z_a$, то тогда цель достигнута, и управление успешно справилось со своей задачей. Вектор z – в классическом понимании – карточка показателей, список индикаторов.

Практика показывает, что в управлении менеджеру необходимо знать не только показатели эффективности управления, которые указывают на то, что цели достигнуты ($z_p = z_a$). Для эффективной работы менеджеру нужно максимально возможное количество информации. У менеджера должны быть в распоряжении так называемые данные о «погоде» в организации, параметры, которые нельзя изменить, но которые важны для управления различными видами деятельности. Таким образом, n -мерное пространство состояний

связано с объектом управления, которое предоставляет данную дополнительную информацию. Здесь y_e – вектор ожидаемых данных о «погоде» в организации, а y_a – вектор актуальных данных о «погоде» в организации. Решение управленческой задачи может быть намного сложнее, если $y_e \neq y_a$, чем это было бы при условии, что $y_e = y_a$. Несмотря на это, достижение управленческих целей ($z_p = z_a$) возможно даже в случае неблагоприятного внешнего положения дел ($y_e \neq y_a$).

В крупных организациях менеджеры, отвечающие за показатели эффективности управления, являются специальными экспертами, которые выполняют данную предварительную работу. В более мелких организациях каждый менеджер должен сам отвечать за показатели эффективности управления. В целом показатели эффективности управления являются специальным аспектом управления.

Иерархия индикаторов. Система измерения показателей эффективности управления часто понимается как пирамида индикаторов, которые имеют между собой взаимосвязи. В итоге контролируется только одно измерение, представленное верхним показателем прибыли.

Сбалансированная система показателей повышения эффективности управления организацией должна отражать желаемую многомерность управления. Однако невозможно полностью согласиться с тем фактом, что точка зрения рассмотрения финансовой стороны, потребителей, процессов, обучения и роста является оптимальным выбором.

Контрольные индикаторы (переменные, показывающие состояние объекта управления) – индикаторы, значения которых количественно показывают приближение к цели, их плановые/целевые значения должны быть определены. Процесс управления сводится к приближению плановых значений к фактическим.

Информационные индикаторы (переменные, показывающие изменения в неуправляемых средах, – как правило, во внешней среде) – индикаторы, значения которых количественно показывают

состояние внешней среды по отношению к объекту управления среды. Управление не может изменить среду, но должно адаптировать объект управления.

Для эффективного управления необходимо решать следующие задачи:

- определить «правильные» параметры для описания объекта управления и его соответствующего окружения;
- найти значения параметров, которые наилучшим образом описывают поставленные цели и ожидаемую ситуацию внешнего положения дел, которую необходимо будет учесть для достижения целей;
- убедиться, что актуальные значения параметров подлежат измерению в текущем или периодичном режимах (данный процесс называется мониторингом);
- регистрировать отклонения и представлять результаты структуре управления;
- помогать структуре управления интерпретировать результаты и принимать «правильные» решения.

Если рассмотреть окружение, процессы и объекты / субъекты взаимодействия, то можно построить схему, приведенную на рис. 9, которая описывает составляющие процесса принятия решений (если рассматривать его как цифровую копию).

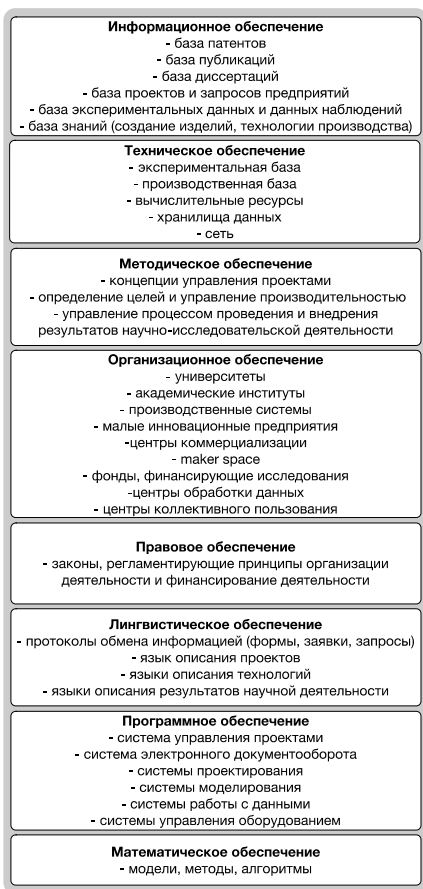
Модель IPOO. Структура индикаторов может быть построена и на основе модели IPOO, где данная аббревиатура обозначает ввод (Input), обработку (Processing), вывод (Output) и результат (Outcome). Такая структура индикаторов имеет строгую ориентацию на ввод и вывод, а также разделяет вывод и результат. Такой подход, однако, не рассматривает проекты, и отобранные индикаторы не обладают достаточной степенью точности данных, которые позволят принять решение о переходе на уровень текущей операции.

Процессный подход. Ещё один способ построения иерархии индикаторов – это рассмотрение только процессов, которые протекают в организационной системе, не затрагивая отдельные проекты, которые могут в ней реализовываться.

СРЕДА РЕАЛИЗАЦИИ ПРОЕКТОВ (ОГРАНИЧЕНИЯ/АКСИОМЫ)

1.

ПОСЛЕДОВАТЕЛЬНОСТИ/ИЕРАРХИЧЕСКАЯ СТРУКТУРА ПРИНЯТИЯ РЕШЕНИЙ



2.

ПРОЦЕССЫ (СВЯЗИ/ОТНОШЕНИЯ) В ЗАДАННОЙ



3.

ОБЪЕКТЫ/СУБЪЕКТЫ ВЗАИМОДЕЙСТВИЯ (КОНЦЕПТЫ)

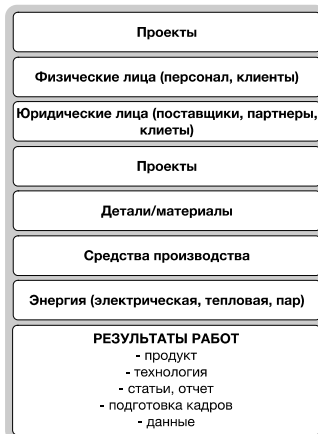


Рис. 9. Информационное окружение производственных систем

В рамках данного подхода индикаторы структурируются по двум составляем группам признаков табл. 6.

Организационные системы, как правило, имеют иерархическую структуру, которая определяется процессом принятия решений, сформировавшимся в рассматриваемой системе (рис. 10).

Таблица 6

Структурирование индикаторов при использовании
процессного подхода

Группа	Классификационный признак 2	Классификационный признак 1					
		1) Простой проект	2) Проект с выделенными этапами	3) Портфель проектов	4) Процессы с выделенными этапами	5) Процессы с размытыми этапами	6) Группа процессов
Изменяемая	A) Выход	A1	A2	A3	A4	A5	A6
	B) Вход	B1	B2	B3	B4	B5	B6
	C) Загрузка	C1	C2	C3	C4	C5	C6
	D) Длительность	D1	D2	D3	D4	D5	D6
	E) График	E1	E2	E3	E4	E5	E6
	F) Доступность	F1	F2	F3	F4	F5	F6
	G) Надежность	G1	G2	G3	G4	G5	G6
	H) Удовлетворенность	H1	H2	H3	H4	H5	H6
Неизменяемая	I) Изменения	I1	I2	I3	I4	I5	I6
	J) Проблемы	J1	J2	J3	J4	J5	J6
	K) Риски	K1	K2	K3	K4	K5	K6
	L) Инциденты	L1	L2	L3	L4	L5	L6
	M) Размер	M1	M2	M3	M4	M5	M6
	N) Сложность	N1	N2	N3	N4	N5	N6

Наличие некоторой иерархии (горизонтальной или вертикальной) приводит к появлению нескольких элементов, принимающих решения (каждый на своем уровне), что при неправильном выборе индикаторов будет приводить к нарушению баланса. Например, при постановке задачи отделу тестирования софтверной компании и выборе в качестве индикатора типы найденных ошибок и их количество мы можем столкнуться с лавинообразным увеличением ошибок, а не повышением качества выпускаемого программного продукта. В отдельных случаях такой показатель может приводить даже к появлению коррупционной составляющей, проявляющейся в договоренности с отделом разработки, который, как правило, получает заработную плату за отработанные часы.

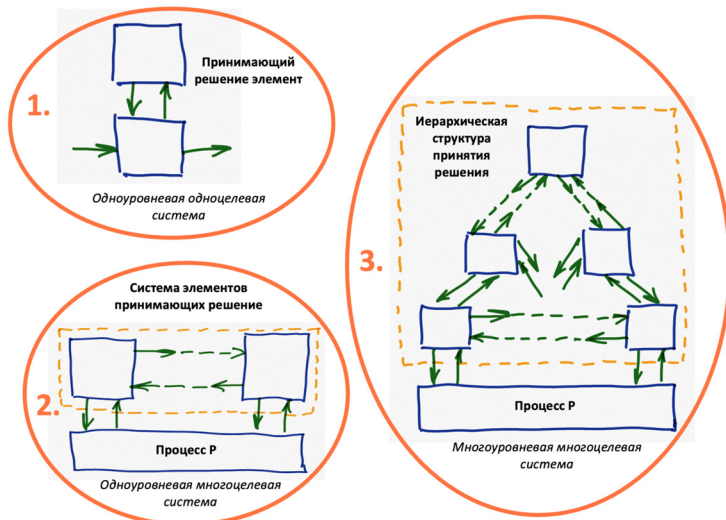


Рис. 10. Многоуровневая многоцелевая система

Построение сбалансированной системы показателей является не тривиальной задачей, решение которой невозможно без должного опыта. При этом существуют рекомендации, которые могут упростить решение этой задачи. К таким рекомендациям следует отнести:

1) выделение типичных сегментов для индикаторов:

- клиенты или пользователи;
- необходимые услуги;
- организация и обеспечение деятельности, процессы;
- служащие;
- финансы;
- принципы управления, риски, безопасность;
- инновации и улучшения;
- поставщики и закупаемые услуги;

2) использование на сегмент не более пяти индикаторов и не более 20 всего.

При построении системы индикаторов использование именно приведенных сегментов и/или всех сегментов является не обязательным. Пример выбора сегментов и показателей приведен на рис. 11.

Финансы Затраты на рабочее место. Степень выполнения бюджетов проектов. Доля закупок по рамочным контрактам. Доля затрат на ИТ в общей доле затрат	Обеспечение деятельности Возраст используемого оборудования. Возраст используемого ПО. Длительность решения проблем служащих. Доступность материалов, инструментов, оборудования, ПО. Доля критически важных ресурсов и процессов
Клиенты Удовлетворенность клиентов предоставляемыми сервисами. Удовлетворенность клиентов предоставляемыми услугами. Степень проникновения решений компании	Организация деятельности Частота и длительность совещаний Качество принимаемых решений Доля бизнес-ориентированных проектов Инновации и улучшения Покрытие потребностей в ноу-хау через обучение Наличие нововведений в деятельности

Рис. 11. Пример сбалансированного набора показателей для организации, работающей в области ИТ

Для использования KPIs необходимо: 1) выбрать «видение» / «глобальные» цели; 2) разработать «глобальную» стратегию; 3) определить промежуточные цели и стратегии / области управления; 4) определить и расставить приоритеты индикаторов (KPIs); 5) определить способ измерения и расчета фактических значений KPIs.

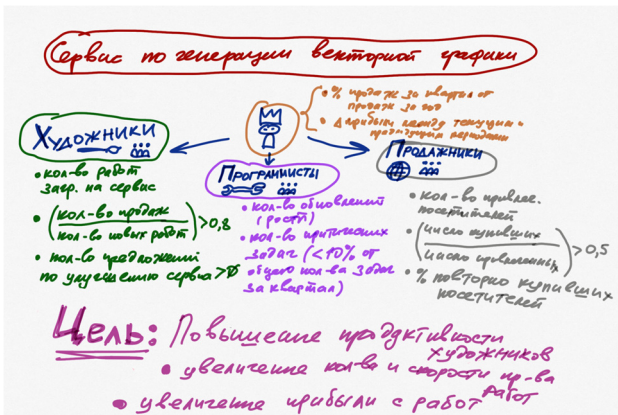


Рис. 12. Пример иерархии показателей, ориентированных на достижение цели

При выборе и работе с КРІ также необходимо обеспечить адекватность и своевременность измерения значений.

Пример цели и индикаторов, направленных на её достижение, приведены на рис. 12.

1.3. Оценка отклонений

Оценка качества планирования и процесса движения к цели может быть осуществлена путем контроля значений показателей в выбранных точках во времени (рис. 13).

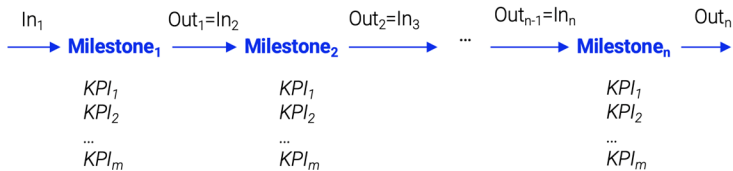


Рис. 13. Последовательность точек контроля показателей

Если исключить из рассмотрения горизонт планирования в явном виде, то уровни управления (такие как стратегическое, тактическое и оперативное) будут различаться величиной отклонений на области поиска решений $\{NPV, NR, R\}$ (рис. 14), что хорошо согласуется с используемым иерархическим принципом планирования – стратегический уровень \Rightarrow тактический уровень \Rightarrow оперативный уровень. Согласно этим уровням в ПрС разделяются полномочия и наблюдается разная частотность планирования на новые этапы.

Если рассмотреть объект управления как черный ящик $B(S, R, T)$ (рис. 15; для оценки субъекта управления потребуется 5–20 непротиворечивых показателей), где $S = (s_1, s_2, \dots, s_N)$ – выходной вектор, $R = (r_1, r_2, \dots, r_M)$ – входной вектор, $T = [t_0, t_n]$ – временной интервал с точками контроля состояния t_0, t_1, \dots, t_n , тогда выходной вектор формируется значениями функции $s_j(t)$, где $s_j(t) \leq s_j(\tilde{t})$ для $t \leq \tilde{t}$ и $1 \leq j \leq N$, а также $s_j(t_0) = 0$. Значениями

входного вектора являются функции $r_k(t)$, где $r_k(t) \leq r_k(\tilde{t})$ для $t \leq \tilde{t}$ и $1 \leq k \leq M$, также $r_k(t_0) = 0$.

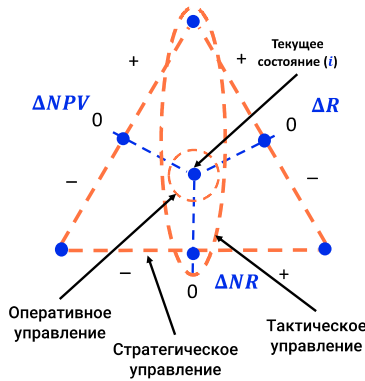


Рис. 14. Область принятия решений при стратегическом, тактическом и оперативном управлении

Задача управления при такой постановке включает запланированные выходы $S_p(t_i)$ и запланированные входы $R_p(t_i)$, а также соответствующие фактические выходы $S_a(t_i)$ и соответствующие фактические входы $R_p(t_i)$.

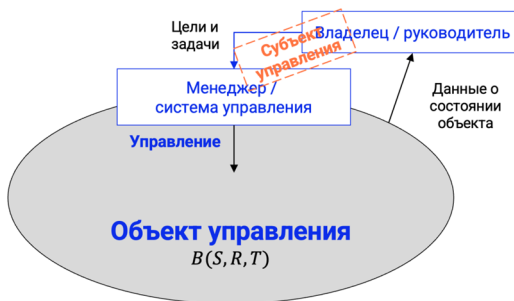


Рис. 15. Схематическое представление взаимосвязи между объектом и субъектом управления

При **оценке отклонений** проверяются значения целевых выходных индикаторов, и они будут достигнуты, если расстояние $d_s(s_a(t_i); s_p(t_i)) \leq \varepsilon_p$, а целевые входные значения будут достигнуты, если расстояние $d_r(r_a(t_i); r_p(t_i)) \leq \varepsilon_r$, где ε – величина допустимой погрешности (в идеальном случае $\varepsilon = 0$), d – метрика, для которой верно, что $d \geq 0$, $d(x, x) = 0$, $d(x, y) = d(y, x)$ и $d(x, z) \leq d(x, y) + d(y, z)$, если $x \leq y \leq z$ (считаем, что $x \geq y$, если $x_1 \geq y_1$, $x_2 \geq y_2$, и т. д.).

В качестве альтернативы будем считать, что цель достигнута, если $s_a(t_i) \geq s_p(t_i)$ или $r_a(t_i) \leq r_p(t_i)$.

Другой способ – это оценивать эффективность. Пусть задано значение функции v_s или v_r , для которых выполняется $v_s \geq 0$, $v_r \geq 0$; $v_s(0) = 0$, $v_r(0) = 0$; $v_s(x) \geq v_s(y)$, $v_r(x) \geq v_r(y)$, если $x \geq y$.

Тогда эффективностью для момента времени t_i будем называть $E(t_i) = \frac{v_s(s(t_i))}{v_r(r(t_i))}$.

Эффективность может быть нормирована некоторым значением $E(B_0)$. Тогда получим $E^* = \frac{E}{E(B_0)}$. $E^*(B_0) = 1$. B_1 более эффективна, чем B_0 , если $s_1 \geq s_0$ и $r_1 \leq r_0$.

Метод анализа среды функционирования (Data Envelopment Analysis – **DEA**) основывается на идеях Фаррелла [31]; метод был описан в работе «Измерение эффективности единиц принятия решений» Чарнсом, Купером и Родсом [32] и впервые применялся при рассмотрении задач линейного программирования для оценки эмпирических технологических границ возможностей ПрС. В Германии аналогичный подход использовался и до выхода указанной работы для оценки предельной производительности НИОКР.

Рассмотрим n сценариев развития, имеющих несколько выходов и несколько входов. Выберем один из этих сценариев. Скон-

струируем новый объект таким образом, чтобы он был линейной комбинацией всех объектов и имел выходные значения выбранного бизнес-объекта при наименьшей взвешенной сумме входов. Может оказаться, что только выбранный объект отвечает этим требованиям. Прделаем это для каждого из n заданных бизнес-объектов. В результате задача выбора стратегии сведется к решению задачи:

$$\max \frac{\sum_{m=1}^M s_m(t_i) \cdot u_m(t_i)}{\sum_{n=1}^N r_n(t_i) \cdot q_n(t_i)},$$

где u и q – весовые коэффициенты. Получаемые значения будут меньше или равны единице, а значения коэффициентов и входных и выходных значений должны быть положительными.

В результате может быть определено количество шагов, которое необходимо сделать системе, чтобы достичь своих конечных целевых значений.

Это входно-ориентированная версия DEA. Иногда рассматривают выходно-ориентированную версию (когда мы ищем максимальный выход по заданному входу).

Для монотонно возрастающих выходных значений мы имеем $S(t_{i+1}) - S(t_i) \geq 0$. Для монотонно возрастающих входных значений мы имеем $R(t_{i+1}) - R(t_i) \geq 0$.

В практической деятельности достижение значений индикаторов больших, чем было запланировано, стоит рассматривать как растрату ресурсов. Тогда эффективный выход $S_u(t_i) = \min(S_a(t_i), S_l(t_i))$, где S_l – ограничение выходного значения.

Аналогично и неиспользованное входное значение стоит рассматривать как растрату ресурсов. Тогда вход $R_u(t_i) = \max(R_a(t_i), R_l(t_i))$, где R_l – предельное (максимально возможное) значение входа.

Пределы выхода или входа имеют аналогии в теории удовлетворения из теории принятия решений. **Удовлетворение** – это стратегия принятия решений, или когнитивная эвристика, которая предполагает

ет поиск доступных альтернатив до тех пор, пока не будет достигнут порог приемлемости. Термин «удовлетворенность» представляет здесь собой сочетание «удовлетворенности» и «достаточности».

Оценка отклонений может осуществляться **со взвешенными суммами**. Рассмотрим $v(s(t)) = \sum_{j=1}^N v v_j \cdot v_j(s(t))$, где $v v_j \geq 0$ – весовые коэффициенты, v_j – значение функции.

Рассмотрим $w(r(t)) = \sum_{k=1}^M w w_k \cdot w_k(r_k(t))$, где $w w_k \geq 0$ – весовые коэффициенты, w_k – значение функции. Необходимо выбрать такие $v_j(s_j(t)) = s_j(t)$ и $w_k(r_k(t)) = r_k(t)$, где v и w – линейные операторы, при которых достигаются локальные уровни целей:

$$t_{loc}(i+1) = \left[v(s_a(t_{i+1})) - v(s_a(t_i)) \right] \cdot \left[v(s_p(t_{i+1})) - v(s_a(t_i)) \right]^{-1},$$

$$u_{loc}(i+1) = \left[w(r_a(t_{i+1})) - w(r_a(t_i)) \right] \cdot \left[w(r_p(t_{i+1})) - w(r_a(t_i)) \right]^{-1}$$

и глобальные цели:

$$t_{glo}(i+1) = v(s_a(t_{i+1})) \cdot v(s_p(t_{i+1}))^{-1},$$

$$u_{glo}(i+1) = w(r_a(t_{i+1})) \cdot w(r_p(t_{i+1}))^{-1}.$$

Если мы выбираем $v_j(s_j(t)) = s_j(t)$ и $w_k(r_k(t)) = r_k(t)$, тогда $v(s(t)) = \text{const}$ и $w(r(t)) = \text{const}$ описывают $(N-1)$ -мерную гиперплоскость в «мерном» выходном пространстве и $(M-1)$ -мерную гиперплоскость в M -мерном входном пространстве.

С использованием функций v и w получим значение локальной эффективности:

$$E_{loc}(t_{i+1}) = \left[v(s(t_{i+1})) - v(s(t_i)) \right] \cdot \left[w(r(t_{i+1})) - w(r(t_i)) \right]^{-1},$$

и значение глобальной эффективности:

$$E_{glo}(t_{i+1}) = v(s(t_{i+1})) \cdot w(r(t_{i+1}))^{-1}.$$

Мы можем рассмотреть текущую эффективность E_a или целевую эффективность E_p путем замены s на s_a или, соответственно, s_p и r на r_a или, соответственно, r_p .

Как уже рассматривали ранее, мы можем нормировать эффективность некоторым значением эффективности E_0 как некоторым рекомендуемым состоянием. Мы можем нормировать текущую эффективность целевым значением.

Такие подходы со взвешенными суммами допускают, что одно и то же выходное значение может быть получено бесконечно большим числом вариантов, а также то, что различные входные значения могут меняться, не оказывая влияния на результат (выходные значения).

Верно ли это для реальных организаций? Часто выходные значения могут быть достигнуты только при определенной комбинации входа (например, при производстве продукции необходима определенная рецептура, количество материалов).

Если $v(s)=s$, то использование взвешенных сумм говорит о том, что значение всегда увеличивается на vv , если добавляется одна единица s (не уменьшается предельная полезность).

Но как же мы определяем значения весов vv_j и ww_k ? Часто эти веса интерпретируются как единичные цены (число ресурса на единицу продукции и т.п.). Если мы рассматриваем входные данные, то это приводит к интерпретации взвешенной входной суммы как затрат.

Каждая система функционирует в некоторой среде. Например, если мы рассматриваем продукцию, то её нельзя рассматривать без рынков, на которых генерируются цены. Мы можем рассчитать себестоимость единицы продукции. Но тогда выходная величина будет идентична входной величине (а эффективность всегда будет

равна «1»). Таким образом может сложиться ситуация, когда мы будем производить ненужную выходную величину или тратить впустую входную величину.

Если веса не являются ни ценами, ни затратами на единицу продукции, то они являются более или менее абстрактными числами эквивалентности для сравнения размера или величины выходных единиц. При таком подходе часто эти веса невозможно измерить напрямую. Поэтому используются специальные методы (например, подход, основанный на сравнении альтернатив, как в методе анализа иерархий).

Рассмотрим обобщение, при котором взвешенная сумма $v(s(t)) \approx \sum_{g=1}^G v v_g \cdot v_g(s_{r_g}(t))$, когда $v v_g \geq 0$ и $v_g(s_{r_g}) \geq 0$, а также $w(r(t)) \approx \sum_{h=1}^H w w_h \cdot w_h(r_{r_h}(t))$, когда $w w_h \geq 0$ и $w_h(r_{r_h}) \geq 0$. Выходные и входные значения аппроксимируются линейной комбинацией выходных и входных векторов (s_{r_g} и r_{r_h}) с известными значениями. Тогда задача сводится к поиску оптимальных значений $v v_g$ и $w w_h$.

Таким образом, мы приходим к оптимизационной задаче:

$$d_s(s(t); \sum_{g=1}^G v v_g \cdot s_{r_g}(t)) \rightarrow \min,$$

$$d_r(r(t); \sum_{h=1}^H w w_h \cdot r_{r_h}(t)) \rightarrow \min.$$

Для хозяйственной деятельности имеет значение только результат (значения выхода), так как вход всегда может быть измерен через затраты $K(r(t))$.

Решение задачи позволяет производить поиск оптимальных решений, когда $s_{app} \geq s$, поиск оптимального решения, когда $s_{app} \leq s$, и др.

Существует несколько вариантов **геометрического подхода** к оценке отклонений. Один их вариантов связан с **оценкой расстояний**. Рассмотрим метрику d_s в «-мерном» пространстве вывода и метрику d_r в M -мерном пространстве ввода, например, метрику Евклида².

Тогда значение функции (метрики) – это расстояние между выбранной начальной точкой и выбранной конечной точкой.

Уровень достижения локальной цели можно в этом случае оценить по формулам:

$$t_{loc}(t_{i+1}) = \frac{d_s(s_a(t_{i+1}); s_a(t_i))}{d_s(s_p(t_{i+1}); s_a(t_i))},$$

$$u_{loc}(t_{i+1}) = \frac{d_r(r_a(t_{i+1}); r_a(t_i))}{d_r(r_p(t_{i+1}); r_a(t_i))},$$

а глобальной цели – по формулам:

²Для оценки величины отклонения значения от заданного наиболее широкое распространение получили меры близости расстояний [7]: евклидово расстояние (расстояние по прямой): $L_2(y, y^*) = \sqrt{\sum_{i=1}^m (y_i - y_i^*)^2}$, где y_i и y_i^* – значения сопоставляемых наборов данных; расстояние Минковского, метрика L_1 , манхэттенское расстояние (расстояние городских кварталов): $L_H(y, y^*) = \sum_{i=1}^m |y_i - y_i^*|$; расстояние Чербышева (метрика шахматной доски): $L_\infty(y, y^*) = \max_{1 \leq i \leq m} |y_i - y_i^*|$.

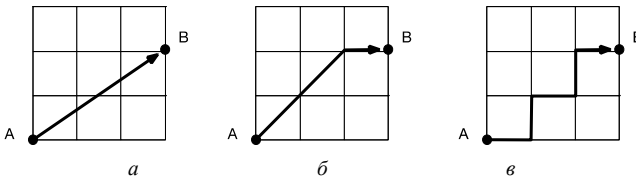


Рис. Демонстрация разницы между принципами вычисления мер близости:
 а – евклидово расстояние; б – расстояние Чербышева; в – манхэттонское расстояние

$$t_{glo}(t_{i+1}) = \frac{d_s(s_a(t_{i+1}); s_a(t_0))}{d_s(s_p(t_{i+1}); s_a(t_0))},$$

$$u_{glo}(t_{i+1}) = \frac{d_r(r_a(t_{i+1}); r_a(t_0))}{d_r(r_p(t_{i+1}); r_a(t_0))},$$

где $s_a(t_0) = s_p(t_0)$ и $r_a(t_0) = r_p(t_0)$.

Уровень достижения цели может быть больше «1» (или 100 %). Для вектора целевых значений это может быть хорошо, однако для вектора входных значений – это всегда нежелательный исход.

Используя концепцию ограниченного выхода, целевой уровень достижения для вектора выхода может быть ограничен «1» и, следовательно, не превышать уровень «1». Тогда такая модель будет иметь особую геометрическую структуру.

При $t(t_{i+1}) = \text{const}$ часть «-мерного» шара (только координаты ≥ 0), и все точки на поверхности этого шара имеют одинаковый уровень достижения цели.

Эффективность может быть оценена по формулам:

$$E_{loc}(t_{i+1}) = \frac{t_{loc}(t_{i+1})}{u_{loc}(t_{i+1})},$$

$$E_{glo}(t_{i+1}) = \frac{t_{glo}(t_{i+1})}{u_{glo}(t_{i+1})}.$$

Нормирование можно осуществить с использованием:

$$E_0 = \frac{d_s(s_p(t_n); s_p(t_0))}{d_r(r_p(t_n); r_p(t_0))}.$$

Альтернативный подход заключается в рассмотрении $s_a(t_i)$ и нахождении вектора $v \cdot s_p(t_n)$ таким, чтобы минимизировать $d_s(s_a(t_i); v \cdot s_p(t_n))$. Это частный случай уже рассмотренного подхода в предыдущем разделе – подхода только с одним опорным век-

тором. Значение уровня достижения цели (v) может превышать единицу ($v > 1$).

Подход допускает изменение точки взгляда, т.е. когда мы наблюдаем процесс достижения цели не из начальной точки, а из целевой точки.

В этом случае

$$t(t_{i+1}) = 1 - \frac{d_s(s_a(t_{i+1}); s_p(t_n))}{d_s(s_p(t_0); s_p(t_n))}.$$

Такая «метрика» может принимать отрицательные значения. Этого можно избежать, если использовать концепцию ограниченного выхода и входа, как указывалось выше.

К концу раздела в рамках рассмотренной концепции мы получили три подхода, которые на практике будут давать отличающиеся результаты.

Также для оценки отклонений в рамках **геометрического подхода**, кроме оценки расстояний, можно использовать длину пути. Если начальное состояние системы было $s_p(t_0)$ и в текущий момент времени t_i мы достигли состояния системы $s_a(t_i)$, то, чтобы достичь состояния $s_p(t_n)$, необходимо перейти из состояния $s_a(t_i)$ в состояние $s_p(t_n)$. Таким образом, длина всего пути, через который мы должны пройти, составляет $d_s(s_p(t_0); s_p(t_i)) + d_s(s_p(t_i); s_p(t_n))$.

Тогда

$$t_{glo}(t_i) = \frac{d_s(s_p(t_0); s_a(t_i))}{(d_s(s_p(t_0); s_a(t_i)) + d_s(s_a(t_i); s_p(t_n)))},$$

$$t_{loc}(t_i) = \frac{d_s(s_a(t_{i-1}); s_a(t_i))}{(d_s(s_a(t_{i-1}); s_a(t_i)) + d_s(s_a(t_i); s_p(t_n)))}.$$

Аналогичные оценки можно получить при движении в другую сторону – от конца к началу. При таком контроле процесса дости-

жения цели вмешательство в процесс необходимо, если мы получаем отрицательные значения, или же когда движение к цели замедляется или меньше некоторого значения при оценке значений вектора индикаторов. Для этого мы рассматриваем три частичных пути (движение от начала, достижение цели (смотрим с конца) и эффективность шага). Общая оценка в данном случае становится многокритериальной задачей, и оценка движения становится сопоставлением каждого шага с уже проделанными на основе таких подходов, как TOPSIS и ETC (Estimate to Complete).

1.4. Основы управления процессами и проектами

Проблема управления и планирования в промышленности была сформулирована А. Калмесом в 1904 году как проблема учёта и статистики в фабричном и товарном производстве [33]. Г. Гант в 1916 году предложил систему графиков, которые помогают спланировать взаимозависимые последовательности работ [34], известные в настоящее время как диаграммы Ганта. Ф. Тейлор продвигал принцип узкой специализации и выделил планирование как один из основных элементов организации производственной деятельности [35]. Г. Файоль определил пять основных функций менеджмента (**планирование, организация, мотивация, контроль, координация**), которые легли в основу теории управления ПрС и проектами. В 1925 году У.Э. Шухарт предложил использовать контрольные карты для отображения изменения показателей во времени, развитием которых стал принцип Деминга – Шухарта (циклы PDSA (Plan-Do-Study-Act) и PDCA (Plan-Do-Check-Act)). В 30-е годы XX века был предложен матричный подход для управления проектами в организационных структурах, в 40-е годы К. Исикава предложил принцип исследования наиболее важных причинно-следственных связей для управления ПрС (диаграмма Исикавы / диаграмма «рыбья кость»); в 50-е годы был разработан метод составления расписаний, метод критического пути и стал использоваться системный подход к управлению проектами; в 60-е годы по-

явились сетевые модели управления проектами. С этих пор в научном плане область управления ПрС и проектами можно считать сформированной, и в дальнейшем она развивалась эволюционно [36].

Использование КРІ приводит к реализации принципа управления, приведенного на рис. 16. Однако для его использования необходима формализация протекающих в системе процессов.



Рис. 16. Алгоритм управления системой на основе использования КРІ

Для описания и сбора в рамках единого информационного пространства протекающих в системе процессов используют концепцию Workflow-Management System (WfMS/WMS), которая опирается на использование методов формализации процессов и их объединения с подходом, основанным на проектах в рамках единого потока заданий. В результате мы получаем формализованный производственный / бизнес-процесс, представленный в виде сети из набора диаграмм (рис. 17) [9].

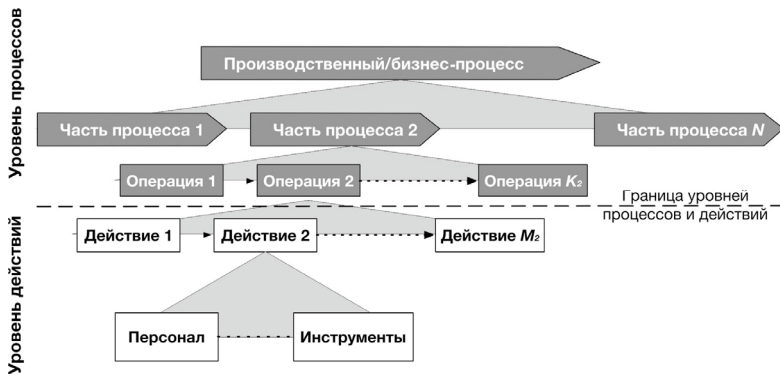


Рис. 17. Структура, описывающая производственный / бизнес-процесс по концепции WfMS

Внедрение концепции WfMS опирается на функции управления, координации, планирования и контроля производственных / бизнес-процессов. При этом реализация концепции не может рассматриваться отдельно от других концепций (концепций управления и структурно-функционального моделирования).

1.5. Механизмы повышения эффективности

В конце XIX – начале XX века Вильфредо Парето, изучая распределение доходов итальянских домохозяйств, получил отношение 80/20, из которого следовало, что 20 % домохозяйств имеют 80 % богатств. В дальнейшем это соотношение много раз встречалось в разных сферах, и для описания этого явления были введены такие понятия, как кривая Лоренца и коэффициент Джини. Было показано, что это соотношение верно и к распределению доходов предприятий. При этом 20 % наиболее удачных предприятий не являются константой. Одни предприятия разоряются или сливаются, другие приобретают богатство и попадают в список «избранных». Разоряются и предприятия, которые на разных периодах входят в список «избранных», однако и они борются за повышение эффективности своего функционирования. В таких условиях использование эффективных интеллектуальных методик принятия решений

и управления становится необходимостью, особенно в условиях конкуренции, когда предприятия-конкуренты уже используют такие методы. Уже можно заметить, что вектор развития постепенно начинает смещаться от монопольного права предприятий выбирать то, что они хотят производить, к проектам, которые начинают выбирать, где они будут производиться.

Механизмом особенно быстрого развития предприятия является выпуск инновационной продукции. При этом неудачное внедрение инновационных проектов может привести к разорению при неэффективной организации производства. Это связано с тем, что рынок, на котором работают современные производственные системы, а особенно ориентированные на выпуск инновационной продукции, является чрезвычайно динамичным. Это говорит о том, что производство такой продукции требует синхронизации выбора выпускаемой продукции с динамикой изменения потребительского спроса, предъявляет повышенные требования к скорости принятия решений и их качеству. Кроме этого, инновационная продукция, как правило, обладает более коротким жизненным циклом, большим количеством модификаций (можно наблюдать переход к мелкосерийному или штучному производству под нужды заказчика), большей наукоемкостью и/или конструктивной сложностью, требует большего количества средств для организации производства и повышенной организационной и технологической гибкости производственных систем. В связи с этим эффективность управления и ошибки, связанные с управлением и планированием такими производственными системами, обретают эффект, непосредственно влияющий на их жизнеспособность [37].

Таким образом, можно определить три фактора, которые оказывают непосредственное влияние на эффективность производственной системы: 1) множество проектов, из которого формируется портфель проектов, которые выпускает производственная система, из чего вытекает задача формирования и управления таким портфелем; 2) синхронизация объема выпуска с потоком заказов; 3) оптимизация процесса организации производства под потребности рынка с учетом особенностей выпускаемого портфеля продуктов.

Для управления реализацией проектов необходимо рассматривать процессы управления, их развитие на разных этапах и переходы от одного этапа на другой (см. рис. 1), который может быть сформулирован как множество переходов $G_i^{(D)} \cup J_i^{(C)} \exists i$:

$$S_{i-3}^{(A)} \cup S_{i-3}^{(B)} \xrightarrow{I_{i-3}^{(B)} \cup I_{i-3}^{(AB)}, P_i^{(B)}, R_{i-3}^{(B)}} S_{i-2}^{(B)} \cup S_{i-2}^{(C)} \xrightarrow{I_i^{(C)} \cup I_{i-2}^{(BC)}, P_{i-2}^{(C)}, R_{i-2}^{(C)}} S_{i-1}^{(C)} \cup S_{i-1}^{(D)} \xrightarrow{I_{i-1}^{(D)} \cup I_{i-1}^{(CD)}, P_{i-1}^{(D)}, R_{i-1}^{(D)}} S_i^{(D)} \cup S_i^{(E)} \xrightarrow{I_i^{(E)} \cup I_i^{(DE)}, P_i^{(E)}, R_i^{(E)}} S_{i+1}^{(E)} \cup S_{i+1}^{(F)} \xrightarrow{I_{i+1}^{(F)} \cup I_{i+1}^{(EF)}, P_{i+1}^{(F)}, R_{i+1}^{(F)}} S_{i+2}^{(F)},$$

где i – номер шага планирования ($n - 1 > i > 2$), T_n – горизонт планирования, I – множество ресурсов (инвестиций), необходимое для осуществления перехода, R – оценка риска, связанная с осуществлением перехода, P – потенциальная прибыль (выгоды), ожидаемая от реализации перехода, S – множество возможных состояний.

Решение задачи в указанной постановке предполагает наличие активного элемента в виде субъекта управления. При этом степень формализации разных этапов, проходимых проектом, различна; поэтому при решении задачи речь не может идти о применении какого-то одного метода или способа, а может идти только о применении группы методологий или подходов в рамках единой теории или концепции.

На сегодняшний день исследователи рассматривают, как правило, переходы в рамках одного этапа. Наиболее разработанным этапом при этом является этап реализации проектов в рамках существующей производственной системы ($S_{i-1}^{(D)} \xrightarrow{I_{i-1}^{(D)}, P_{i-1}^{(D)}, R_{i-1}^{(D)}} S_i^{(D)}$).

Возможность приведенного разбиения задачи на составляющие говорит, с одной стороны, о возможности ее решения и использования для этого существующих подходов для решения отдельных составляющих, с другой стороны, о необходимости учета такого фактора, как фактора времени при планировании и управлении, а также других факторов, влияющих на реализуемость принимаемого решения и выступающих его ограничениями: технические и технологические возможности реализации проекта и его выпуска как товара в соответствии с поступающими заказами; финансовые возможности

в плане необходимых оборотных средств, средств для необходимой модернизации технологической и производственной базы; необходимые компетентные возможности сотрудников предприятия.

Рассматривая в качестве внешней среды множество рынков и проектов, следует учитывать высокий уровень вариативности и динамики протекающих процессов вкупе с инертностью производственных систем, что накладывает ограничения на применимость методов, на которых могут базироваться теории и концепции управления производственными системами. Например, использование таких подходов, как управление по фактическим данным, рефлексивное управление, целевое управление без учета динамики изменения среды, в которой функционирует производственная система, приводит к задержке в принятии и применении управленческих решений, что проявляется в ошибках управления, приводящих к несогласованности действий подсистем и нарушениях производственных циклов во времени.

Принципы управления, используемые лицами, принимающими решения, могут базироваться на разных стратегиях поведения. При взаимодействии производственных систем с рынком и инновационными проектами они могут искать оптимальные для себя стратегии, исходя из сложившейся или складывающейся (по прогнозным оценкам) ситуации, а могут заниматься выработкой воздействий на внешнее окружение, которые приведут к желаемому для конкретной производственной системы состоянию.

Рассмотрение процесса появления и выхода на рынок инновационных проектов через их реализацию в производственных системах можно представить схемой, приведенной на рис. 18, где A – множество идей, B – множество инновационных проектов, C – множество производственных проектов, D – множество производственных систем, E – множество сбытовых систем, F – множество рынков.

Учитывая сложность рассматриваемых процессов, актуальным становится выбор способа их описания и исследования для принятия управленческих решений.

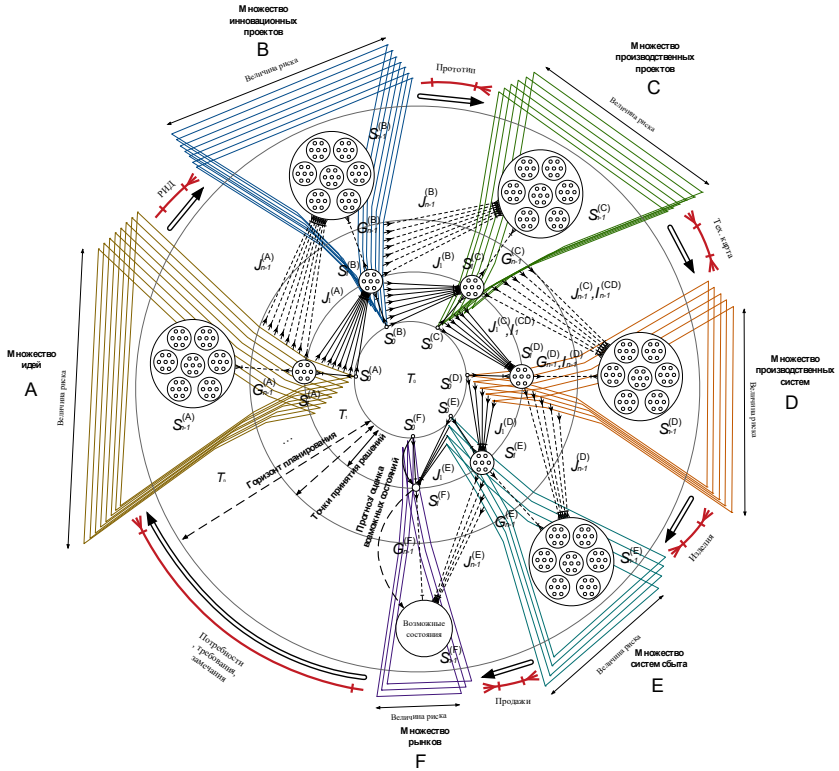


Рис. 18. Структура взаимодействия элементов организационной системы с выделенными моментами времени (шагами / тактами контроля)

Так как описываемый процесс затрагивает множество сфер деятельности, то невозможно говорить о реализации некоторой системы поддержки принятия решений, а необходимо говорить об информационной инфраструктуре, которая должна состоять из средств для «вычисления решений», основанных на использовании моделей как ряде процедур по обработке данных и суждений, помогающих ЛПР в принятии решений [38].

При разработке информационной системы речь должна идти прежде всего, о механизме работы с данными и способах интеграции компонентов модели в рамках единого информационного пространства [39].

Глава 2. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ*

Реальность бесконечно сложна
для нашего познания. Мы должны
упрощать.

Олдос Хаксли

Согласно отчёту Grand View Research [40], размер мирового рынка управления бизнес-процессами оценивался в 14,46 млрд долларов США в 2022 году, и ожидается, что совокупный годовой темп роста (CAGR) составит 19,9 % в период с 2023 по 2030 год. Этот рост можно объяснить растущей потребностью организаций в оптимизации своей деятельности и совершенствовании бизнес-процессов.

Развитие информационных технологий привело к появлению целых классов специализированных систем – *MES, PLM, EAM, MRP, MRP II, CIM, CALS, ERP, ERP II, CRM, SCM, PRM*, которые собирают и обрабатывают большое количество информации.

В настоящее время разработка новых информационных решений – это формализованный процесс. Сегодня при создании информационной инфраструктуры организационных систем выделяют три уровня: онтологий, метаданных и уровень модели [41]. На уровне онтологий – это категории и события, участвующие в процессах организационной системы. На уровне метаданных оперируют информацией, которая характеризует процессы; для разработки решений, основанных на данных, существуют такие модели, как *SEMMA* [42], *CRISP* [43], *ASUME*. На уровне моделей для описания протекающих процессов реализуются структурно-функциональные модели, опирающиеся на концепции *WfMS* [9] и *Business Process Re-engineering (BPR)* [44]. Для структурно-функционального описания модели необходимо выстраивать соотношения между используе-

* Глава написана Л.А. Мыльниковым, Пермский национальный исследовательский политехнический университет.

мыми методами, опираясь на соответствие между их входами и выходами для формирования процессов (последовательностей действий), согласно целям функционирования организационных систем. Формализация структурно-функциональных моделей требует применения нотаций eEPC [8], *BPMN* [45], *IDEF* или их символьных представлений [46].

Развитие методов структурно-функционального моделирования связано с расширением спектра учитываемых факторов. Концепция *TOGAF* и её реализация – язык *ArchiMate* [11] позволяют рассматривать во взаимосвязи структурно-функциональные модели, средства их реализации и цели функционирования организационных систем. Несмотря на то, что субъект может быть связан с описываемыми процессами и системой мотивации, данная методология не допускает исследования степени влияния этого субъекта на эффективность процессов организационной системы.

Оценка возможностей предлагаемых информационных инфраструктур вызывает необходимость использования либо алгоритмических описаний, позволяющих устанавливать взаимосвязи между графическим представлением информационной инфраструктуры и её функционалом [47] (например, *BPEL*), либо мнений экспертов при построении нескольких уровней большей детализации процессов [48] (такой подход свойственен концепциям *SADT* [49] и *ARIS*). Существуют исследования, в которых изучаются вопросы влияния точности собираемых данных на работу информационной инфраструктуры [50], работы алгоритмов, а также времени выполнения тех или иных операций [51]. Однако приведенные здесь методологии структурно-функционального моделирования не позволяют оценивать влияние субъекта управления на эффективность инфраструктуры поддержки принятия решений, а также, несмотря на то что такие подходы повышают эффективность протекающих бизнес-процессов, они не делают возможным проведение сравнения структурно-функциональных моделей по степени их эффективности до реализации этих моделей в организационной системе.

Влияние субъекта на эффективность работы информационной инфраструктуры можно учитывать статистическими методами при

использовании методов структурно-событийного моделирования (например, на основе сети Петри и *YAWL* [52]). Однако это не способствует поиску эффективных алгоритмов взаимодействия субъекта с информационной инфраструктурой, а также не позволяет учитывать влияние других внешних для исследуемого процесса факторов, имеющих стохастическую природу.

Для повышения эффективности организационных систем за счет изменения информационной инфраструктуры применяются хорошо зарекомендовавшие себя решения (базы решений ИТІЛ и СОБИТ), типовые паттерны [53], обратный инжиниринг протекающих в системах процессов на основе собираемых логов [54] (процесс майнинга для поиска несоответствий между закладываемыми процессами и фактическими действиями) и семантической информации [55]. Развитие этих подходов некоторые авторы видят в стандартизации бизнес-процессов по принципу снизу вверх или сверху вниз [53].

2.1. Взаимосвязь эффективности и организации процессов

Понимая под информационной инфраструктурой совокупность моделей, методов, алгоритмов обработки информации, аппаратного обеспечения и способ их использования (последовательность применения/структуру), мы можем представить её в обобщенном виде как функцию $Sys = F(S, I, A, O, R)$, структуру взаимосвязи которой можно представить, как показано на рис. 19.

Детализация системы может осуществляться либо во времени, тогда в динамике система будет описываться множеством состояний:

$$Sys(t_1) \rightarrow Sys(t_2) \rightarrow \dots \rightarrow Sys(t_n),$$

где n – количество рассматриваемых моментов времени.

Также система может детализироваться в глубину, когда действия раскрываются на последовательное или параллельное выполнение операций с соответствующими им блоками S, I, O, R ,

$$Sys = \bigcup_i Sys_i,$$

где каждый элемент может быть также детализирован до необходимого уровня.

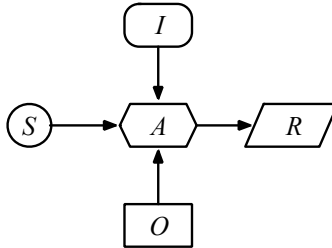


Рис. 19. Схематичное представление информационной инфраструктуры (*S (Subject)* – субъект управления; *A (Action)* – действие, операция, метод, алгоритм; *I (Input)* – входные данные, окружение; *O (Object)* – объект, на котором выполняются действия (технические средства); *R (Result)* – результат)

Элементы системы могут быть классифицированы по источнику оказываемого на них воздействия. Все элементы являются зависимыми от внутренних решений. В дополнение к этому элемент *I* является также зависимым от окружения системы.

От качества принимаемых решений будет зависеть эффективность организационной системы (*EF*), для чего необходимо сформировать группы критериев. Любой выбор будет связан с учетом нескольких факторов ($EF = F(NPV, NR, R)$, где *NPV* – факторы оценки эффективности (экономические, социальные, экологические и др.), *NR* – ресурсы, *R* – риск). Множество критериев, участвующих в управлении, и отклонения от найденных целевых значений, которые мы будем наблюдать в процессе реализации, определяют уровни управления (рис. 20).

Для решения задач многокритериального выбора широкое распространение получили подходы на основе экспертного выбора (метод Кондорсе, выбор большинством, метод Борда, метод с двумя турами, методы анализа иерархий и сетей), теории игр (игра с нуле-

вой суммой), методы ранжирования (Promethee, Vikor, Electre, Topsis) и подходы на основе выбора критерия оценки (группа методов MAUT).

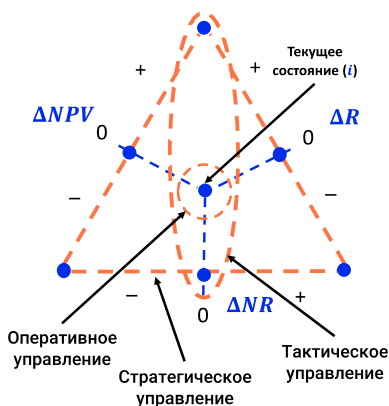


Рис. 20. Область принятия решений при стратегическом, тактическом и оперативном управлении

Рассмотрение информационной инфраструктуры как функции $F(S, A, I, O, R)$, которая определяет структуру взаимодействия элементов, позволяет провести формальную постановку группы задач поиска на заданной структуре взаимодействия.

Задача параметрической оптимизации: $S = ?$, $A = \text{const}$, $I = \text{const}$, $O = \text{const}$, $R = \text{const}$. Определение параметров работы элементов S , приводящих к максимальной эффективности (периодичность вызова тех или иных методов), – $\text{argmax}_S EF$. Примером оп-

тимизируемых параметров могут быть периоды оперативного, тактического и стратегического планирования / перепланирования. Для решения этой задачи необходима имитация деятельности всей системы для оценки рассматриваемых вариантов.

Задача выбора методов: $S = \text{const}$, $A = ?$, $I = \text{const}$, $O = \text{const}$, $R = \text{const}$. Если выбор осуществляется из ограниченного списка методов, имеющих соответствие входов и выходов под заданную структуру, то задача может быть решена перебором. При неопреде-

ленной структуре множество рассматриваемых методов приводит к увеличению размерности задачи структурного поиска за счет увеличения размерности на величину, равную мощности множества методов.

Задача определения оптимальных условий окружения (формирования требований к окружению): $S = \text{const}$, $A = \text{const}$, $I = ?$, $O = \text{const}$, $R = \text{const}$. Данная задача рассматривается как определение оптимального набора внутренних параметров и прогнозирования внешних факторов. Как правило, данная задача не рассматривается в отдельности, а выступает в качестве ограничений.

Задача определения требований к технической базе: $S = \text{const}$, $A = \text{const}$, $I = \text{const}$, $O = ?$, $R = \text{const}$. Если используемым в процессе принятия решений методам сопоставить в соответствие вычислительную сложность, то задача формирования требований может быть приведена в соответствие с требованиями к вычислительным ресурсам, исходя из требований производительности под заданную нагрузку.

Задача поиска целевого значения результата (задача планирования) – $S = \text{const}$, $A = \text{const}$, $I = \text{const}$, $O = \text{const}$, $R = ?$ К этой группе задач следует отнести задачу объемно-календарного планирования [57].

Рассматривая информационную инфраструктуру, мы сталкиваемся с ситуацией, когда в процессе реального управления все из перечисленных задач необходимо решать. С их решением связаны вопросы модернизации производства, смены номенклатуры изделий, заказчиков и поставщиков, изменения технологических процессов.

Параллельное или последовательное применение метода / выполнение операций соответствует вариантам графа с максимальной длиной $l \in [1, 2, 3, \dots, n]$, где n – число элементов действия A ($l = 1$ соответствует конфигурации, когда все действия выполняются параллельно, а n – когда все действия выполняются последовательно). При этом для $l = 1$ мы имеем 1 возможный вариант; $l = 2$ будет $2 \cdot 2! = 4$ вариантов; для $l = 3$, $3 \cdot 3! + 2 \cdot 2! = 23$; $4 \cdot 3! + 4$; $l = 5$,

$$\sum_{i=0}^5 (n-i) \cdot (n-i)! = 719; \text{ для } l = n, \sum_{i=0}^{n-1} (n-i) \cdot (n-i)! \text{ Таким образом,}$$

мы наблюдаем резкий рост вариантов по мере детализации и/или увеличения масштаба задачи (табл. 7). Из таблицы можно сделать вывод, что задачу невозможно решить за приемлемое время перебором даже возможных структур без учета инвариантности методов, так как реальных процессов, описывающих работу предприятий, сотни и тысячи операций в зависимости от степени детализации.

Таблица 7

Количество комбинаций для разного количества элементов действия и необходимое время, если принять, что одна комбинация оценивается в течение 1 секунды

1	1	2	3	4	5	10	15
Количество вариантов	1	4	6	119	719	39 916 799	20 922 789 887 999
Время в минутах	<1	<1	<1	~2	~12	~665280	~348713164800
Время в днях	<1	<1	<1	<1	<1	462	242161920

В общем же виде задача может быть поставлена как $\text{argmax}_{\text{Sys}} EF$.

Решение поиска структуры и параметров, приводящих к оптимальной работе организационной системы, является NP-трудной задачей и для частных случаев (применительно к производственным системам) рассматривалась Ж. Тиролем [15].

На сегодняшний день отсутствуют универсальные механизмы формализации и решения задачи в общем виде. В работе была предложена новая нотация, позволяющая производить моделирование работы информационной инфраструктуры. Таким образом, можно получать данные, позволяющие оценивать работу исследуемой системы. Однако вопрос генерации вариантов возможных решений оказывается не раскрытым.

2.2. Моделирование информационных процессов с выделенным субъектом управления

Повышение эффективности (EF) инфраструктуры поддержки принятия решений сложных организационных и технических систем (контура обработки информации) связано с эффективностью используемых методов ($EF^{(M)}$), методики их использования ($EF^{(USE)}$), организации их взаимодействия (структурно-функциональная эффективность – $EF^{(STRUCT)}$) и обеспечивающей их работоспособности информационных решений ($EF^{(INF)}$):

$$EF = f\left(EF^{(M)}, EF^{(USE)}, EF^{(STRUCT)}, EF^{(INF)}\right).$$

Каждую из этих составляющих стоит рассматривать как переменную величину, состоящую из затрат на модернизацию (разовые затраты на перевод системы в желаемое состояние) и затрат на эксплуатацию (стоимость решения задач / выполнения операций).

Поиск оптимального решения $EF \rightarrow opt$ невозможен, так как неизвестно все множество методов (M), способов структурно-функциональной организации ($STRUCT$) и возможных решений обеспечения деятельности информационной инфраструктуры (INF). Различные методы могут включать в себя разное количество решаемых задач; для множеств методов может быть эффективна разная структурная организация и требоваться различная техническая поддержка для обеспечения деятельности информационной инфраструктуры. Поэтому стоит рассматривать поиск лучшего решения на ограниченном множестве методов, методик использования, структур, решений, обеспечивающих функционирование информационной инфраструктуры.

Качество решения задачи выбора эффективной инфраструктуры поддержки принятия решений связано с возможностью генерации множества решений (SOL) и сравнения их эффективности (EF).

Сравнивать возможные решения необходимо с точки зрения целей функционирования организационной системы (см., например,

постановку цели по SMART [28]), в чьих интересах реализуется информационная система поддержки принятия решений. Для оценки степени достижения целей вводится множество KPI (см., например, пирамиду PMS, IPOO-модель, метод баланса показателей GRC [17]).

Работа информационной инфраструктуры зависит от множества используемых методов, характеризующихся свойствами (вычислительной сложности (V), множества входных (NV) и выходных (R) данных), появления управляющих событий (S), которые вносят инвариантность решений за счет разных состояний окружения (NV и R) и характеризующейся периодичностью их появления, а также алгоритмической реализации методов (V).

Для алгоритмов, алгоритмическая сложность которых неизвестна, величина запаздывания получения результата в зависимости от входных данных требует отдельного исследования [58] и может быть измерена для уже реализованных подсистем и алгоритмов [59].

Выбор эффективной комбинации методов (M), способов структурно-функциональной организации ($STRUCT$) и возможных решений обеспечения деятельности информационной инфраструктуры (INF) подразумевает перебор вариантов их сочетаний, что требует соответствующего инструментария, позволяющего оценивать и выбирать наилучшие решения следующей задачи:

$$SOL = \underset{S, V, NV, R, USE, M, INF, STRUCT}{\operatorname{argmax}} EF, \quad (1)$$

где $STRUCT = O, H$ – категория, в которой O – множество объектов $K = \{K_M, K_{USE}, K_{INF}\}$, а H – множество морфизмов $H = \{H_S, H_V, H_{NV}, H_R\}$ [60].

В результате приведенных здесь рассуждений задача повышения информационной инфраструктуры для поддержки принятия решений может быть представлена двумя представлениями (рис. 21) [61].

На основе представления, приведенного на рис. 2, б, введем двойственную категорию $STRUCT^{(2)} = O^{(2)}, H^{(2)}$, в которой O – множество объектов $K = \{K_S, K_V, K_{NV}, K_R\}$, а H – множество мор-

физмов $H = \{H_M, H_{USE}, H_{INF}\}$. При этом существующие нотации не обладают всеми необходимыми типами объектов K и позволяют реализовать только один тип морфизмов (H_{USE}).

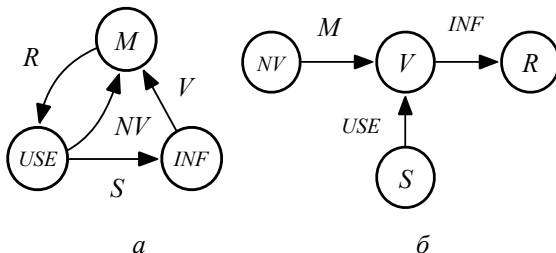


Рис. 21. Прямая и двойственная постановки задачи: *a* – через выбор метода решения, методики использования и реализации информационной системы; *б* – через выбор реализации метода (алгоритма), практики применения, набора необходимых данных [29]

Обеспечение производственно-информационной системы через информационные решения опирается не только на потребность периодического использования методов (M) в условиях ограниченного времени на получение результата, но и зависит от вычислительной сложности используемых алгоритмов. Для разных методов имеются разные реализации, которые могут существенно снизить производительность используемого метода. На общую производительность информационной инфраструктуры для поддержки принятия решений будут оказывать влияние объем и частота передаваемых данных, а на выбор методов, методику работы и структуру – стоимость требуемой для реализации инфраструктуры (внедрения и обслуживания).

В настоящее время не существует инструментальных средств, позволяющих находить оптимальное решение (SOL) с учетом всех составляющих, оказывающих влияние на его эффективность.

Ни один из существующих подходов не позволяет методами численных оценок оценивать эффективность и сравнивать предлагаемые решения. Такое сравнение требует выделения субъектов управления в явном виде, использования средств имитационного


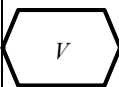


моделирования, введения системы внутренних тактов для ухода от реального времени и привязки к производительности системы, на которой производится имитация работы и накопление статистики при изучении влияния случайных факторов. Анализ их эффективности в существующих системах имитационного моделирования затруднен тем, что эти системы имеют свои наборы структурных элементов и типов связей.




Способы представления имитационных моделей не связаны напрямую с существующими концепциями структурно-функционального моделирования, что делает актуальной разработку множества артефактов, которые могут найти прямое отражение в системах имитационного моделирования и реализовать подходы к исследованию существующих и предлагаемых способов формализации процессов поддержки принятия решений [27].

Для описания информационной инфраструктуры при решении задач поддержки принятия решений введем графические обозначения K и H (табл. 8).

Таблица 8

Обозначение узлов / блоков для структурно-функционального моделирования информационной инфраструктуры и взаимосвязей между ними

Обозначение	Описание
	Блок управления (S) – выдает бинарный сигнал 0/1 по некоторому заданному внутри блока алгоритму (1 – означает запуск алгоритма, реализованного в последующем блоке)
	Блок действия (V) – реализует операцию, результатом которой будут данные, передаваемые в блок результата (R), значения переменных, необходимых для работы блока; берутся из блока Контекста (NV).
	Результат (R) – блок, хранящий результаты, получаемые при реализации блока процесса (V).
	Блок контекста (NV) – хранит данные, необходимые для работы блока Действия (V).

Обозначение	Описание
	Блок условия. Используется для ожидания выполнения действий (синхронизации и управления последовательностью процессов)
	Поток данных
	Поток событий (передает значения 0/1)

С использованием введенных обозначений соответствующая информационной инфраструктуре категория может быть представлена схемой, приведённой на рис. 22.

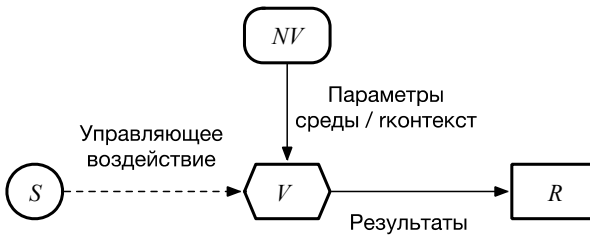


Рис. 22. Структурно-функциональное представление работы информационной инфраструктуры при решении задач поддержки принятия решений

Введем символическое обозначение работы системы: $V(NV, S) = R$. Усложнение (детализация) системы может осуществляться путем параллельного или последовательного раскрытия блока операции/действия (V) при сохранении равенства правой и левой частей выражения.

Предположение 1 (о детализации блоков действия). Блоки действия (V) могут быть представлены последовательностью более простых операций / действий с внутренними промежуточными результатами и параметрами среды. Такая детализация может осуществляться путем параллельного и последовательного рас-

крытия элементов модели (рис. 23), где некоторые элементы могут не выделяться (например, $S1=S2$ на рис. 23, а), а также отдельные элементы S , NV , R могут быть окружением детализируемого элемента. При этом параллельному и последовательному раскрытиям соответствуют символичные формы записи:

$$1) \text{ последовательное раскрытие: } V(NV, S) = V2(NV2, S2) = \\ = V2(V1(NV, S), S2) = R;$$

$$2) \text{ параллельное раскрытие: } V(NV, S) = V1(NV, S) \cup V2 \times \\ \times (NV, S) = R.$$

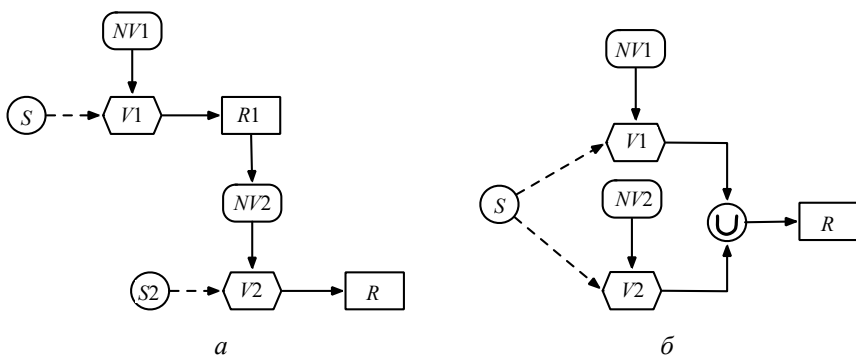


Рис. 23. Способы структурной детализации: а – для последовательных задач, б – для параллельных задач (такие блоки, как объединение/пересечение, условный блок и другие вспомогательные блоки могут быть реализованы через введение блока операций. Таким образом, введенный набор множества типов объектов (K) и морфизмов (H) является достаточным)

Предположение 2 (о детализации блоков действия). Операция объединения результатов (рис. 23, б) может быть заменена блоком операции/действия (рис. 24), что соответствует символическому представлению:

$$V(NV, S) = V1(NV, S) \cup V2(NV, S) = \\ = V3((V1(NV, S), V2(NV, S)), S) = R.$$

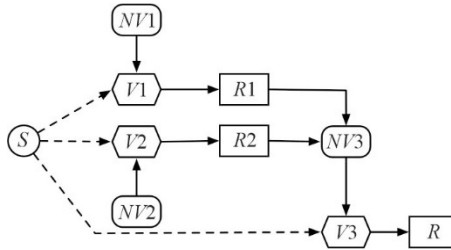


Рис. 24. Способы структурной детализации операции объединения результатов работы параллельных блоков

Предположение 3 (о детализации блоков действия). *Операция условия может быть заменена блоком операции / действия (рис. 25), что соответствует символическому представлению:*

$$V(NV, S) = V2(NV2, S2) = V2(V1(NV, S), S2) = R.$$

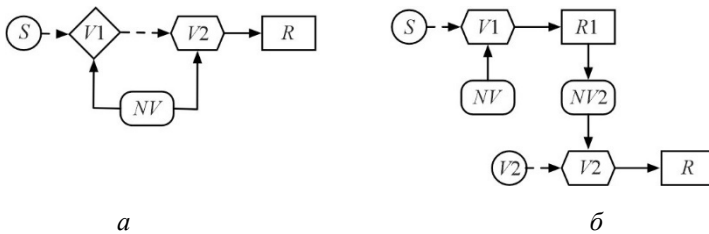


Рис. 25. Способы (а, б) структурной детализации операции условия

Таким образом, имея множество пар «действие – результат» (V, R) и опираясь на правила замены, может быть сконструировано множество конфигураций и требований.

Для функционирования информационной инфраструктуры необходимо сформулировать **требования по производительности** для операции ограничения на максимальное значение амплитуды $V(NV, S)$:

$$\max \left[\frac{1}{v(S)} \right] < T(O^{V(NV)}(N(S))) \quad \forall S, V,$$

где $v(S)$ – частота; T – время выполнения операции сложности O (для операции $V(NV)$ при окружении NV); N – число воздействий при управлении S .

Для оценки времени работы всей информационной инфраструктуры введем способ вычисления общего времени работы для детализаций:

последовательной – $T = T(O^{V1(NV)}(N(S))) + T(O^{V2(NV^2)}(N(S2)))$
(см. предположение 1);

параллельной – $T = \max(T(O^{V1(NV)}(N(S))), T(O^{V2(NV)}(N(S)))) + T(O^{V3(NV^3)}(N(S)))$ (см. предположение 3);

условной – $T = T(O^{V1(NV)}(N(S))) + T(O^{V2(NV^2)}(N(S2)))$ (см. предположение 3).

Время работы информационной инфраструктуры до получения конечного результата R будет определять величину запаздывания, которая, в свою очередь, зависит от числа возмущающих воздействий и горизонта планирования.

Получаемая схема может быть смоделирована, если задать начальные значения модели (NV_0), описать алгоритмы работы или принять характеристики блоков действий (V) и принципы работы элементов управления (S), связанные как с моментами времени появления процессов, протекающих в системе (например, оперативный, тактический и стратегический уровни планирования), так и во внешней среде (сбыт, появление новых проектов и др.).

Предположение 4 (о взаимосвязи РРІ и КРІ). *Характеристики информационной инфраструктуры (показатели производительности процессов – PPI) $S, V, NV, R, USE, M, INF, STRUCT$ оказывают влияние на показатели эффективности организационной системы (KPI).*

Исходя из предположения 4, наличие данных о показателях оценки эффективности организационной системы позволяет методами имитационного моделирования выполнять анализ влияния

производительности информационной инфраструктуры на эффективность системы в целом.

Чтобы отделить получаемые результаты от производительности вычислительной системы, на которой выполняется моделирование, введем систему шагов / тактов расчета (i) – каждый из них может иметь различные значения соответствия реальному времени. Для обратного пересчета необходима функция $t(i, p)$, где p – производительность системы. Значения функции зависят от особенностей реализации алгоритмов, используемой аппаратной и программной базы, что будет приводить к увеличению или уменьшению времени реализации отдельных операций. При таком подходе время становится интервально-заданной величиной.

При рассмотрении влияния случайных факторов (например, появления новых проектов), согласно центральной предельной теореме [62], необходимы множественные расчеты.

Использование средств моделирования информационной поддержки принятия решений с помощью информационных систем позволяет автоматизировать процессы повышения эффективности, создания эффективной информационной инфраструктуры и поиска решений в ходе функционирования организационной системы для достижения заложенных целевых показателей.

Алгоритм поиска решения может быть представлен следующим образом:

Шаг 1. Построение модели, описывающей существующую или предполагаемую информационную инфраструктуру $A^{(0)}$.

Шаг 2. Генерация множества вариантов альтернативных структур и категорий для них. Запись их в множество вариантов A .

Шаг 3. Параметризация вариантов из множества A (начальное состояния NV , управление S и способ реализации I) и запись полученных вариантов во множество B .

Шаг 4. Определение воздействий, меняющихся случайным образом, и числа необходимых экспериментов для учета этого фактора.

Шаг 5. Формирование требований и ограничений по производительности алгоритмов и аппаратного обеспечения.

Шаг 6. Оценка эффективности использования вариантов из множества B .

Шаг 7. Если вариант решения из множества B не может быть реализован без модернизации на существующей базе, то необходимо сформировать требования к её изменению, оценить затраты и перейти на шаг 5, иначе – добавить рассматриваемый вариант решения в множество выбора C .

Шаг 8. Произвести ранжирование вариантов на основе КРІ одним из методов многокритериального выбора.

Шаг 9. Если найден приемлемый вариант организации информационной инфраструктуры (решения), то перейти на шаг 14.

Шаг 10. Сократить множество C до N лучших вариантов.

Шаг 11. $B = C$.

Шаг 12. Создать новые варианты и добавить их в множество B :
– при соответствии структур произвести попарный обмен способами параметризации (операция кроссовера);
– изменить структуры путем обмена их частями при соответствии входов и выходов отдельных элементов.

Шаг 13. Перейти на шаг 5.

Шаг 14. Конец работы алгоритма.

Вопрос генерации вариантов структурно-функциональной организации информационной инфраструктуры следует рассматривать как вопрос, связанный с фиксацией входа, выхода и окружения. Для этого сложилась практика AS-IS- и TO-BE-анализа процессов компании.

В качестве примера рассмотрим задачу объемно-календарного планирования производственной деятельности, представленной в [63]. Перейдя к буквенным обозначениям блоков, получим схему (структурно-функциональную модель), приведенную на рис. 26.

Для проведения численных расчетов на структурно-функциональной модели (имитационного моделирования работы информационной системы поддержки принятия решений) необходима информация о том, как будет осуществляться управление.

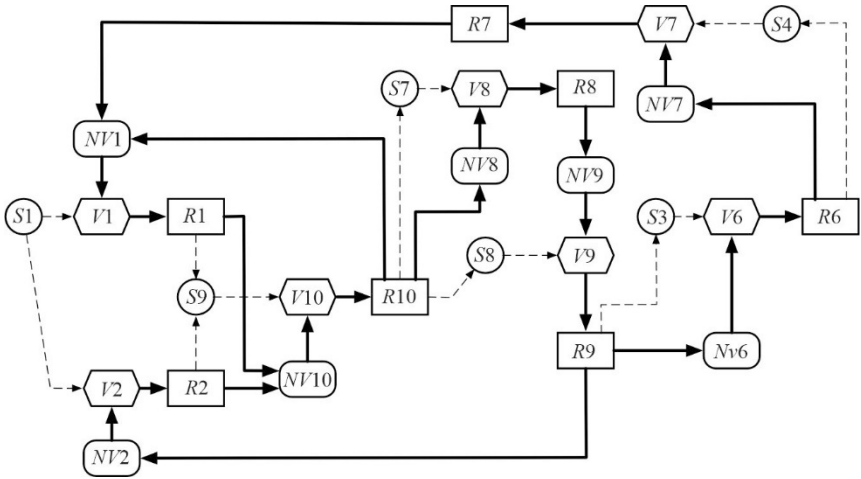


Рис. 26. Структурно-функциональная модель управления портфелями проектов производственной системы: **S1** – появление нового проекта, **S3** – производство, **S4** – время сбыта/продажи, **S7** – изменение набора или состава возможных производственных корзин, **S8** – объемно-календарное планирование, **S9** – проверка изменений, **V1** – исключение проектов из производственной корзины, **V2** – включение проектов в производственную корзину, **V6** – производство, **V7** – сбыт/продажа, **V8** – формирование множества производственных корзин, **V9** – объемно-календарное планирование, **V10** – проверка на наличие проектов, меняющих активную производственную корзину, **R1** – множество проектов на исключение из производственной корзины, **R2** – множество проектов на включение в производственную корзину, **R6** – товары и изделия, **R7** – объемы сбыта/продаж, **R8** – возможные варианты производственных корзин, **R9** – объемно-календарный план, **R10** – активная производственная корзина, **NV1** – данные о продажах товаров из активной производственной корзины, **NV2** – текущие планы производства и ресурсно-технологическое обеспечение, **NV6** – данные о планах и ресурсах, **NV7** – спрос и состояние склада, **NV8** – активная производственная корзина, **NV9** – возможные варианты производственных корзин, **NV10** – производственная корзина и множество проектов на включение и исключение

Случайную величину представляет собой время появления нового проекта, что может быть промоделировано вероятностью его

появления. Время оперативного планирования – $S5$, планирование тактического уровня – $S8$. Примем, что каждый шаг равен одному часу, $S1$ будем описывать потоком Пуассона с вероятностью появления проектов, равной 0,1; $S9$ будет наступать один раз в год ($365 \cdot 24 = 8760$ часов), $S3$ – раз в 10 дней ($10 \cdot 24 = 240$ часов), $S8$ – раз в квартал (3 месяца) ($31 \cdot 3 \cdot 24 = 2232$ часа), $S7$ наступает по заранее заданному графику (возьмем раз в 10 дней) ($10 \cdot 24 = 240$ часов). Работа алгоритмов описывается разной сложностью: $V1$ – экспоненциальной; $V2$ – линейной; $V10, V8, V9, V6$ – также экспоненциальной. При этом получаемые значения будут не в часах, а в минутах. Предположим, что блоки NV, R отрабатывают за один шаг. Промоделировав работу производственной системы для 10 активных проектов, получим циклограмму, на которой будет видно, что на уровне оперативного планирования система не справляется с нагрузкой.

Сокращение числа активных проектов в производственной корзине до 9 позволяет решить проблему с алгоритмической сложностью приведённой организации информационной инфраструктуры с заданными параметрами.

При увеличении вероятности появления новых проектов до 0,7 наблюдается уже неспособность алгоритмов отбора проектов работать в приемлемое для рассматриваемой информационной инфраструктуры время (рис. 27), так как запаздывание, связанное с поиском решения, превышает интервал цикла управления.

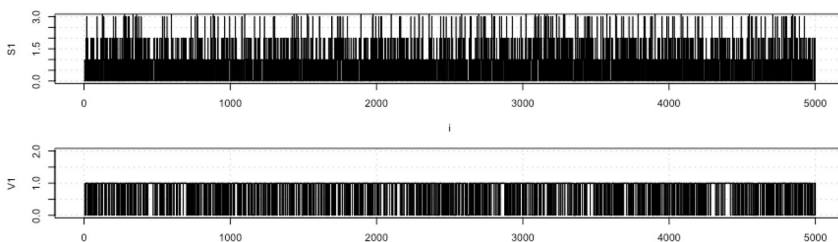


Рис. 27. Циклограмма работы блоков появления новых проектов и исключения проектов из корзины проектов производственной системы модели, приведенной на рис. 26

Определение допустимых параметров, связанных с периодичностью принятия управленческих решений, нагрузок, которые система может выдержать, позволяет перейти к поиску наиболее эффективных решений. В работе [63] приводится анализ экономической эффективности на основе моделирования в условиях, когда время, сопоставляемое для всех шагов, одинаково, модернизация не проводится, объем материалов на складе ресурсов не ограничен.

Для определения зависимости экономической эффективности от величины запаздывания будем использовать следующий алгоритм:

Шаг 1. Задаем количество необходимых повторов для сбора статистики и множество сдвигов на организацию выпуска.

Шаг 2. Задаем количество шагов (i).

Шаг 3. Генерируем моменты появления проектов (поток Пуассона); для каждого проекта – время жизни, функции изменения спроса и цены на основе инновационных кривых для выбранного сдвига выпуска продукции.

Шаг 4. Для каждого i решаем задачу

$$\sum_{j=1}^N p_j(i) * x_j \rightarrow \max,$$

$$x_j \leq s_j(i), \forall j \in [1, N],$$

$$\sum_{j=1}^N y_j \leq Prj,$$

$$x_j \leq s_j(i) * y_j, \forall j \in [1, N],$$

где x – количество проектов в производственной корзине; j – номер проекта; y – бинарный флаг выпуска проекта; Prj – число проектов, которое может выпускаться одновременно; i – шаг расчета; $p(i)$ – прибыль, получаемая на i -м шаге, $s(i)$ – ограничение спроса на i -м шаге, N – число проектов в производственной корзине.

Шаг 5. Вычисляем результаты накопительным итогом.

Шаг 6. Если количество шагов повтора меньше заданного, то переходим на шаг 3, иначе – на шаг 7.

Шаг 7. Вычисление статистики: средних значений прибыли и дисперсии для заданного сдвига.

Шаг 8. Если все сдвиги / количество проектов / горизонты планирования перебраны, то переход на **шаг 9**, иначе – выбор нового сдвига и переход на **шаг 3**.

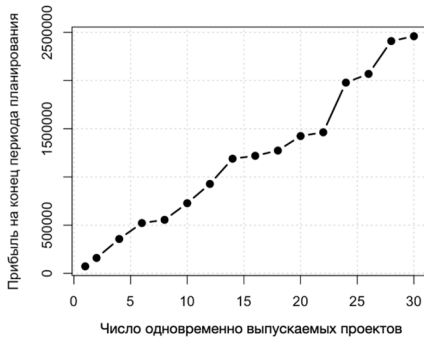
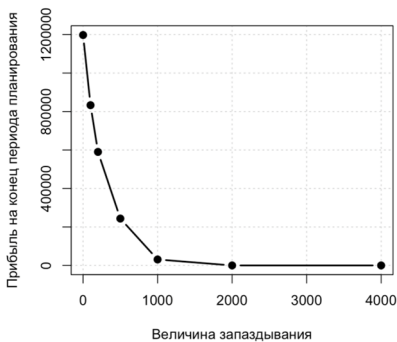
Шаг 9. Окончание работы алгоритма.

При моделировании запаздывания выпуска продукции, описываемого инновационными кривыми [64], стоимость и объем спроса будем уменьшать на величину, соответствующую значению, когда наш проект появляется на рынке (т.е. уменьшенную на величину текущего спроса и стоимости).

Так как характеристики новых проектов и время их появления мы рассматриваем как стохастическую величину, то, согласно методу Монте-Карло [65] и центральной предельной теореме [62], в качестве результатов будем рассматривать усредненные значения характеристик, влияющих на экономическую эффективность производственной системы и получаемых в результате моделирования (рис. 28). Каждая из информационных инфраструктур вместе с производственной системой и сферой деятельности будет описываться своим набором кривых и различаться углом наклона, положением, остротой углов изгиба.

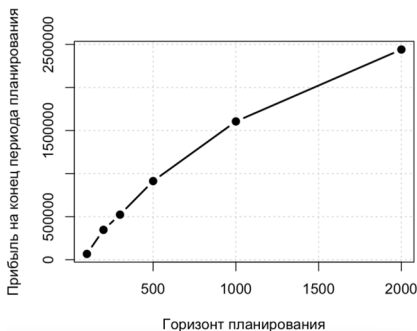
Как видно из графиков, приведённых на рис. 28, эффективность организационной системы зависит от работы информационной инфраструктуры, обеспечивающей поддержку принятия решений, от числа одновременно выпускаемых проектов (X) и выбранных горизонтов планирования, используемых методов, которые описываются некоторыми функциональными зависимостями.

Предположение 5 (о функциональном описании характеристик информационной инфраструктуры). *Зависимость между эффективностью организационной системы и параметрами информационной инфраструктуры системы поддержки принятия решений может быть представлена функциональными зависимостями в аналитическом виде.*



а

б



в

Рис. 28. Оценка экономической эффективности модели, приведенной на рис. 27: *а* – при 500 повторах расчета, среднем времени жизни проекта 1000 единиц, объеме спроса в 5000 единиц, прибыли в 200 единиц за единицу товара, при вероятности появления нового проекта 0,1, горизонте планирования в 500 единиц и 10 проектах, выпускаемых одновременно; *б* – при 500 повторах расчета, среднем времени жизни проекта 1000 единиц, объеме спроса в 5000 единиц, прибыли в 200 единиц за единицу товара, при вероятности появления нового проекта 0,1, горизонте планирования в 500 единиц и запаздывании в 100 единиц; *в* – при 500 повторах расчета, среднем времени жизни проекта 1000 единиц, объеме спроса в 5000 единиц, прибыли в 200 единиц за единицу товара, при вероятности появления нового проекта 0,1, 10 одновременно выпускаемых проектах и запаздывании в 100 единиц

Если в качестве оценки эффективности организационной системы принять прибыль, то, опираясь на полученные графики (см. рис. 28), можно предположить, что методом регрессии могут быть построены модели и найдены значения коэффициентов при выборе следующих математических описаний:

$$EF(i) = \frac{a(i)}{\sqrt{Shift}} \text{ для зависимости } a \text{ на рис. 28,}$$

$$EF(i) = b(i) \cdot N \text{ для зависимости } b \text{ на рис. 28,}$$

$$EF(i) = c(i) \cdot \sqrt{\Gamma} \text{ для зависимости } c \text{ на рис. 28.}$$

Выражая для рассматриваемой конфигурации организационной системы (*STRUCT*) эффективность способов организации управления (*S*) и учитывая, что $EF = \sum_{i=1}^{\Gamma} \left(\sum_{j=1}^N p_j(i) \cdot x_j - Z(i) \right)$, получаем задачу оптимального выбора:

$$\begin{aligned} & \sum_{i=1}^{\Gamma} \left(\sum_{j=1}^N p_j(i) \cdot x_j - Z^{(k)}(i) \right) \rightarrow \max, \\ & \sum_{i=1}^{\Gamma} \left(\frac{a(i)}{\sqrt{Shift(Z^{(k)})}} - Z^{(k)}(i) \right) = \sum_{i=1}^{\Gamma} (b(i) \cdot N - Z^{(k)}(i)), \\ & \sum_{i=1}^{\Gamma} (b(i) \cdot N - Z^{(k)}(i)) = \sum_{i=1}^{\Gamma} (c(i) \cdot \sqrt{\Gamma} - Z^{(k)}(i)), \\ & Z^{(k)} \in \mathbb{Z}, \forall k, \end{aligned}$$

где k – номер варианта по модернизации системы поддержки принятия решений; $Z^{(k)}(i)$ – затраты на внедрение k -варианта на i -м шаге; $Shift(Z^{(k)})$ – величина запаздывания для k -варианта.

В задачах управления портфелями производственных проектов вводится понятие устойчивости [63] как *способность системы со-*

хранять экономическую эффективность при ограниченных отклонениях в интенсивности потока проектов и их характеристиках, связанных с объёмом спроса и цены. Значения минимального и максимального потока проектов и границы времени их эффективного выпуска определяют пределы устойчивости и могут использоваться для введения ограничений в задачах планирования и управления производственной деятельностью. Таким образом, исследование информационной инфраструктуры получает связь с предметной областью.

Использование структурно-функционального представления при наличии информации о вероятности отказа каждого из блоков операции / действия позволяет оценивать надежность всей информационной инфраструктуры [66].

В результате векторный критерий выбора способа организации информационной инфраструктуры из множества возможных решений формируется из показателей, характеризующих информационную инфраструктуру (надежность, величина запаздывания) и показателей предметной области (для рассмотренного примера – это эффективность принимаемых решений, величина возможных отклонений, при которых система остается устойчивой).

Выбор одного из решений приведет к кастомизации информационной инфраструктуры. При этом выбранное решение будет иметь эффективность ниже, чем целевое ее значение за счет ограниченного набора рассматриваемых вариантов. Возникновение такой разницы было объяснено Г. Саймоном в рамках теории ограниченной рациональности [5]. В результате удастся установить взаимосвязь между производительностью работы системы поддержки принятия решений, её экономической эффективностью и временем планирования, что позволяет:

1) перейти от измерения производительности конкретных реализаций [59] к получению функциональных характеристик информационной инфраструктуры;

2) объективно получить значения целевых показателей в контрольных временных точках – $B(OUT, IN, T)$, где $OUT =$

$= (out_1, out_2, \dots, out_N)$ – выходной вектор, $IN = (in_1, in_2, \dots, in_M)$ – входной вектор, $T = [t_0, t_n]$ – временной интервал с точками контроля состояния t_0, t_1, \dots, t_n , соответствующих моментам времени, совпадающим с тактами структурно-функциональной модели;

3) принять решение о внедрении новых методов, алгоритмов, информационных систем, об изменении организационной инфраструктуры на основе данных о затратах на их внедрение и обслуживание и увеличение прибыли, связанной с ростом производительности информационной инфраструктуры и числа внедряемых проектов.

Введение субъекта управления в явном виде позволяет исследовать, с одной стороны, его влияние на работоспособность всей системы в целом (в том числе с учетом используемой методологии), а с другой стороны, проводить обучение на модели в случаях, когда субъект представлен человеком.

2.3. Сравнение и выбор способов реализации информационных процессов

При исследовании информационных процессов наибольшее распространение получили сети Петри благодаря их способности обеспечивать визуальное представление сложных систем и гибкости при моделировании различных типов процессов. Одним из главных преимуществ сетей Петри является их способность моделировать как поток управления, так и потоки данных в системе. Это делает их подходящими для анализа сложных систем, где существует множество взаимозависимых процессов, которые необходимо учитывать [67]. Они особенно полезны при моделировании и анализе сложных систем со множеством взаимодействующих компонентов, поскольку позволяют представлять параллельные действия и имитировать динамическое поведение.

Имея структурно-функциональную модель и информацию о времени выполнения операций, экспертные оценки о вероятности успешности / неуспешности выполнения операций, мы совмещаем

два источника данных: модель и данные, получаемые с объекта управления, и тем самым, согласно [68], данные, полученные после этого методами имитационного моделирования, будут достаточны для поддержки принятия решений об эффективности функционирования системы (рис. 29).



Рис. 29. Шаги исследования структурно-функциональных моделей при их сравнении

Таким образом, сравнивать эффективность становится возможным, только если процесс структурирован (мы имеем дело с так называемыми «лазанья»-процессами) [14].

Рассмотрим пример описания информационного процесса на задаче разработки документации для строительства. Этот процесс является сложным, так как подразумевает совместную работу множества сотрудников, относящихся к разным подразделениям, а также к разным организациям (исполнители и заказчики). Информационные процессы в этом случае связаны с передачей информации между участниками проекта (выдача заданий, справочная информация, информация о согласованных решениях и т.д.). Необходимость держать в голове или обновлять данные о ходе проекта при одновременной реализации множества проектов существенно увеличи-

вает время, необходимое для внесения изменений и дополнений в них в случаях, когда согласование с заказчиком осуществляется только в конце реализации проекта (после прохождения многочисленных внутренних этапов) [69].

В приведенном процессе можно выделить несколько этапов – принятие проекта в работу, сбор исходных данных, принятие инженерных решений согласно техническому заданию, оформление итоговой документации и выдача её заказчику. Результаты, полученные на этапе принятия инженерных решений, проектировщику необходимо согласовывать с руководителем проекта, отвечающим за реализацию проекта в организации исполнителе, а итоговую документацию – как с руководителем проекта внутри организации, так и с заказчиком (рис. 30). Рассматриваемый этап проектирования соответствует модели управления «водопад», которая подразумевает последовательный переход от этапа к этапу и не предполагает возвратов и доработок при выявлении недостатков.

Такая организация является основной сложностью при работе, так как любое замечание заказчика приводит к существенным модификациям результатов, полученных на этапе (затрат времени и объем работ, сопоставимый с затратами на этап). Это связано с тем, что заказчик может повлиять на ход проекта только в самом конце процесса, документация может не пройти согласование у него из-за недостатков, которые можно было бы выявить в ходе реализации этапа. Кроме этого, отсутствует горизонтальное взаимодействие между заказчиком и непосредственным исполнителем, что может приводить к различным взглядам на реализацию проекта, необходимости последующей доработки и согласования позиций. При этом с точки зрения исполнителя, часть процессов на стороне заказчика является необозримой и неуправляемой, поэтому изменения в организации процесса должны давать эффект, независимо от процессов, происходящих на стороне заказчика.

Модель бизнес-процессов с учетом обозначенных недостатков может быть представлена, как приведено на рис. 31.

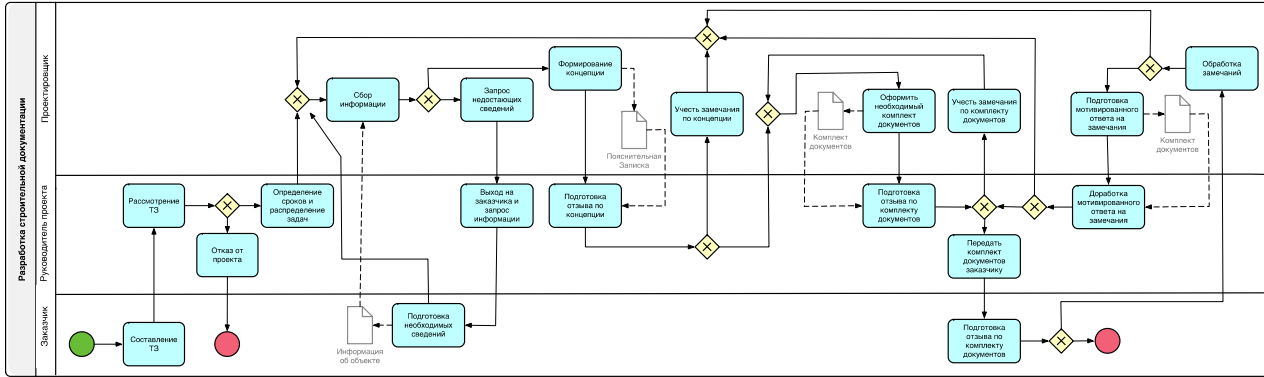


Рис. 30. BPMN-модель рассматриваемого процесса до внесения изменений

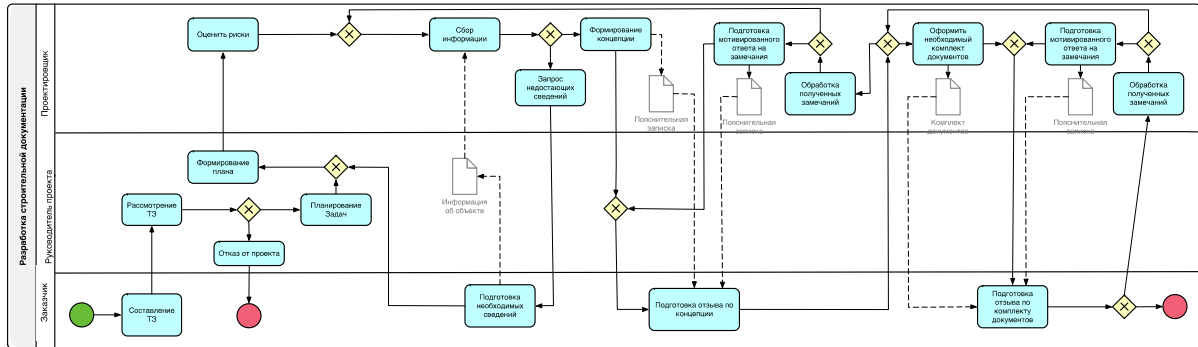


Рис. 31. BPMN-модель предлагаемых изменений в рассматриваемый процесс

Основные операции, выполняемые при работе над проектом, остались те же, однако связь между организациями заказчиком и проектировщиком осуществляется напрямую и происходит в нескольких точках подготовки проектной документации. Параллельно происходит оповещение руководителя проекта на стороне исполнителя, который может скорректировать задачи на основании полученной информации либо вмешаться в процесс взаимодействия при необходимости. При приведённой организации процесса руководитель перестаёт быть посредником, при этом не теряя контроль на всех стадиях (сбор информации, принятие инженерных решений, оформление документации), которые при необходимости могут повторяться по отдельности. Переход между стадиями происходит только при полном согласовании их результатов, что позволяет заказчику добиться от исполнителя наиболее удовлетворительного результата, а проектировщик, в свою очередь, избавляется от лишней работы по оформлению документации.

Порядок взаимодействия рассматриваемого процесса с окружением может быть представлен схемой, приведенной на рис. 32.

Для объективного обоснования замены одной организации процесса на другую необходимы данные численного сопоставления.

Тогда при рассмотрении двух вариантов они должны иметь соответствия по входам и выходам

$$\text{Начало/Вход} \begin{pmatrix} PR_1 \\ PR_2 \\ \dots \end{pmatrix} = \begin{pmatrix} \text{Конец}_1 / \text{Выход}_1 \\ \text{Конец}_2 / \text{Выход}_2 \\ \dots \end{pmatrix},$$

$$\text{Конец}_1 / \text{Выход}_1 = \text{Конец}_2 / \text{Выход}_2 = \dots,$$

где PR – процедурные модели.

Процедурные модели могут быть представлены, например, в форме сетей Петри или структурно-функциональной модели.

Для этого рассмотрим две сети Петри (до изменений и после) [69]. Опишем их как $C = (P, T, I, O)$ и $C^* = (P^*, T^*, I^*, O^*)$, где P

и P^* – множества позиций, T и T^* – множества переходов, I и I^* – входные функции отображения, O и O^* – выходные функции отображения.

$$P = \{p_{\text{нач}}, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_{\text{кон}}, A, B, C, D, E, F, O, U_1, U_2, U_3, V, Z_1, Z_2\},$$

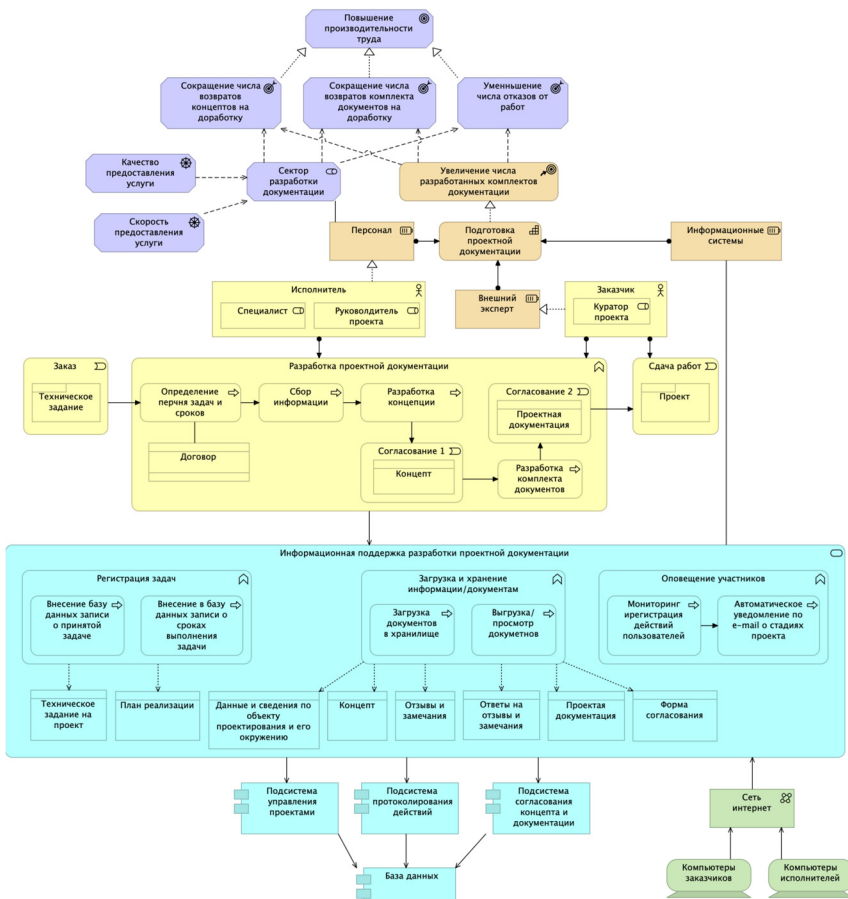


Рис. 32. Схема взаимодействия процесса с его окружением в нотации ArchiMate [11]

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}\},$$

$$I(t_1) = \{p, p_7, U_1, Z_2\} \quad O(t_1) = \{A\}$$

$$I(t_2) = \{A\} \quad O(t_2) = \{Z_1, B\}$$

$$I(t_3) = \{C\} \quad O(t_3) = \{U_1, p_1\}$$

$$I(t_4) = \{p_6, p_1, U_2\} \quad O(t_4) = \{D\}$$

$$I(t_5) = \{D\} \quad O(t_5) = \{F\}$$

$$I(t_6) = \{F\} \quad O(t_6) = \{U_2, p_2\}$$

$$I(t_7) = \{p_2, p_3\} \quad O(t_7) = \{E\}$$

$$I(t_8) = \{E\} \quad O(t_8) = \{p_0, U_3\}$$

$$I(t_9) = \{U_3\} \quad O(t_9) = \{O, p_4\}$$

$$I(t_{10}) = \{O\} \quad O(t_{10}) = \{p_3, p_5\}$$

$$I(t_{11}) = \{p_4, p_5\} \quad O(t_{11}) = \{p_6, p_7\}$$

$$I(t_{12}) = \{Z_1\} \quad O(t_{12}) = \{V\}$$

$$I(t_{13}) = \{V\} \quad O(t_{13}) = \{Z_2\}$$

$$I(t_{14}) = \{B\} \quad O(t_{14}) = \{C\}$$

$$P^* = \{p_{\text{нач}}^*, p_1^*, p_2^*, p_3^*, p_{\text{кон}}^*, A, B, C, D, E, O_1, O_2, U_1, U_2, Z_1, Z_2\},$$

$$T^* = \{t_1^*, t_2^*, t_3^*, t_4^*, t_5^*, t_6^*, t_7^*, t_8^*, t_9^*, t_{10}^*\},$$

$$I^*(t_1^*) = \{p^*, p_2^*, Z_2\} \quad O^*(t_1^*) = \{A\}$$

$$I^*(t_2^*) = \{A\} \quad O^*(t_2^*) = \{B, Z_1\}$$

$$I^*(t_3^*) = \{B, O_1\} \quad O^*(t_3^*) = \{C\}$$

$$I^*(t_4^*) = \{C\} \quad O^*(t_4^*) = \{U_1, p_1^*\}$$

$$\begin{aligned}
 I^*(t_5^*) &= \{p_1^*, p_3^*\} & O^*(t_5^*) &= \{D\} \\
 I^*(t_6^*) &= \{D, O_2\} & O^*(t_6^*) &= \{E\} \\
 I^*(t_7^*) &= \{E\} & O^*(t_7^*) &= \{U_2, p_0^*\} \\
 I^*(t_8^*) &= \{U_1\} & O^*(t_8^*) &= \{p_2^*, O_1\} \\
 I^*(t_9^*) &= \{U_2\} & O^*(t_9^*) &= \{p_3^*, O_2\} \\
 I^*(t_{10}^*) &= \{Z_1\} & O^*(t_{10}^*) &= \{Z_2\}
 \end{aligned}$$

Латинскими буквами (A, B, C, \dots) обозначены позиции, соответствующие одинаковым для обеих сетей процессам, требующим времени на выполнение. Такие операции, как «Подготовка ответа на замечания» (O), «Обработка / учёт замечаний» (U), «Запрос недостающих сведений» (Z), в ходе процесса встречаются несколько раз и с разным функционалом, поэтому соответствующие им в сети Петри позиции отмечены одинаковыми буквами с различными индексами (табл. 9).

Таблица 9

Основные процессы и их характеристики
(k – количество проектов)

Место в сети Петри	Соответствующий процесс	Время работы процесса при одном обращении, ч	Вероятность перехода для схемы до и после изменений, %	
			до	после
A	Сбор информации	8	100	100
B	Формирование концепции	8	50	50
C	Подготовка отзыва	16	100	100
D	Оформление документов	40	100	100
E	Передача документов заказчику	8	100	100
F	Согласование с руководителем	2	100	–

Место в сети Петри	Соответствующий процесс	Время работы процесса при одном обращении, ч	Вероятность перехода для схемы до и после изменений, %	
			до	после
O_1	Подготовка ответа на замечания	4	50	70
O_2			–	70
U_1	Обработка/учёт замечаний	2	50	50
U_2			30	60
U_3			60	–
V	Выход на заказчика	8	100	–
Z_1	Запрос недостающих сведений	8	50	50
Z_2			100	100

Позиции, отмеченные маленькой буквой p с индексом, отражают состояние проекта (промежуточные результаты) и не соответствуют какому-либо реальному процессу, требующему временных затрат, поэтому при проведении экспериментов не учитываются.

Преобразование одного описания в другое возможно только на основании соответствия входов и выходов. Это подтверждается графическими представлениями, приведенными на рис. 30, 31. Такое представление позволяет проводить имитацию прохождения заявки по одной от входа до выхода, определяя её путь в местах развилки согласно указанным вероятностям. Таким образом, путем проведения множественных расчётов может быть собрана статистика для анализа времени обработки заявок при использовании структурно-функциональных моделей обеих структур, обеспечивающих выполнение рассматриваемого процесса. Возможность такого способа исследования подтверждается центральной предельной теоремой (теорема о больших числах) [62], из которой следует, что среднее случайных величин есть величина неслучайная.

Исследование на сетях Петри не позволяет учитывать характеристики внешней среды – такие, как неравномерности появления проектов, а значит исследовать возможность справляться с пропускной способностью и способностью обрабатывать неравномерности появления проектов.

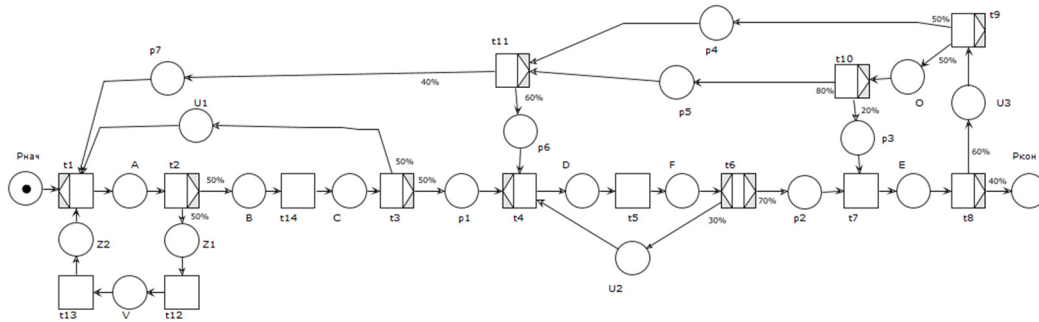


Рис. 33. Сеть Петри, описывающая процесс подготовки строительной документации, до внесения изменений

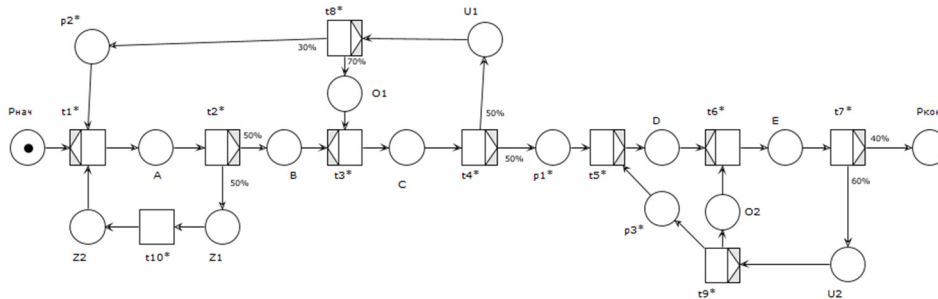


Рис. 34. Сеть Петри, описывающая процесс подготовки строительной документации, после изменений структуры процесса

Для учета этого можно использовать описание с выделенным субъектом управления [70]. Для моделирования условных переходов введем дополнительный условный элемент (рис. 35).

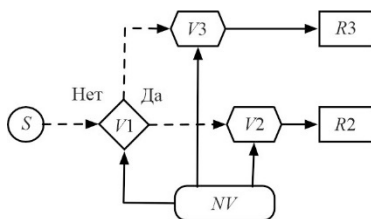


Рис. 35. Обозначение и принцип использования условного элемента

В нотации с выделенным субъектом управления схемы будут выглядеть, как приведено на рис. 33, 34.

Для представления способа до изменений введем обозначения, представленные в табл. 10, и тогда получим схему, приведенную на рис. 36.

Таблица 10

Система обозначений для структурно-функциональной модели процесса подготовки строительной документации до внесения изменений и выявленные соответствия с приведенной на рис. 36 сетью Петри

Обозначение	Соответствия на сети Петри (см. рис. 34)	Время обработки, ч	Сложность	Описание
$S1$	$P_{нач}$	–	–	Поступление нового проекта
$V1$	$P_{нач}$	2	k	Вынесение решения о принятии проекта в работу
$NV1$	$P_{нач}$	–	–	Данные о свободных ресурсах
$R1$	$P_{нач}$	–	–	Множество проектов, принятых в работу
$V2$	A, Z_1, Z_2	18	k	Сбор информации/исходных данных
$NV2$	A	–	–	Техническое задание проекта
$R2$	A	–	–	Исходные данные проекта

Обозначение	Соответствия на сети Петри (см. рис. 35)	Время обработки, ч	Сложность	Описание
$V3$	B	8	k	Формирование концепции проекта
$NV3$	B	–	–	Исходные данные проекта
$R3$	C	–	–	Пояснительная записка
$V4$	C, U_1	18	$\sum k$	Проверка концепции проекта на соответствие требованиям
$NV4$	C	–	–	Концепция проекта
$R4$	P_1, U_1	–	–	Результаты внутреннего согласования 1-го этапа
$NV5$	$p1$	–	–	Внутренне согласованная концепция проекта
$V5$	D	40	k	Оформление документации
$R5$	D	–	–	Комплект документов
$NV6$	D	–	–	Документация
$V6$	F, U_2	4	$\sum k$	Проверка документации проекта на соответствие требованиям
$R6$	P_2, U_2	–	–	Результаты согласования 2-го этапа
$NV7$	P_2	–	–	Внутренне согласованная документация
$S2$	E, U_3	–	–	Реакция заказчика на подготовленные материалы
$V9$	$p_{\text{кон}}$	4	k	Передача материалов заказчику, подписание актов
$R9$	$p_{\text{кон}}$	–	–	Разработанные и сданные заказчику документы
$S3$	U_3	–	–	Реакция исполнителя на замечания
$V10$	O	4	k	Формирование мотивированного ответа на отказ без доработки
$R10$	p_3	–	–	Пакет материалов без доработки с комментариями
$V11$	p_4	2	k	Формирование замечаний и заданий на доработку
$R11$	p_4, p_5	–	–	Пакет материалов для доработки

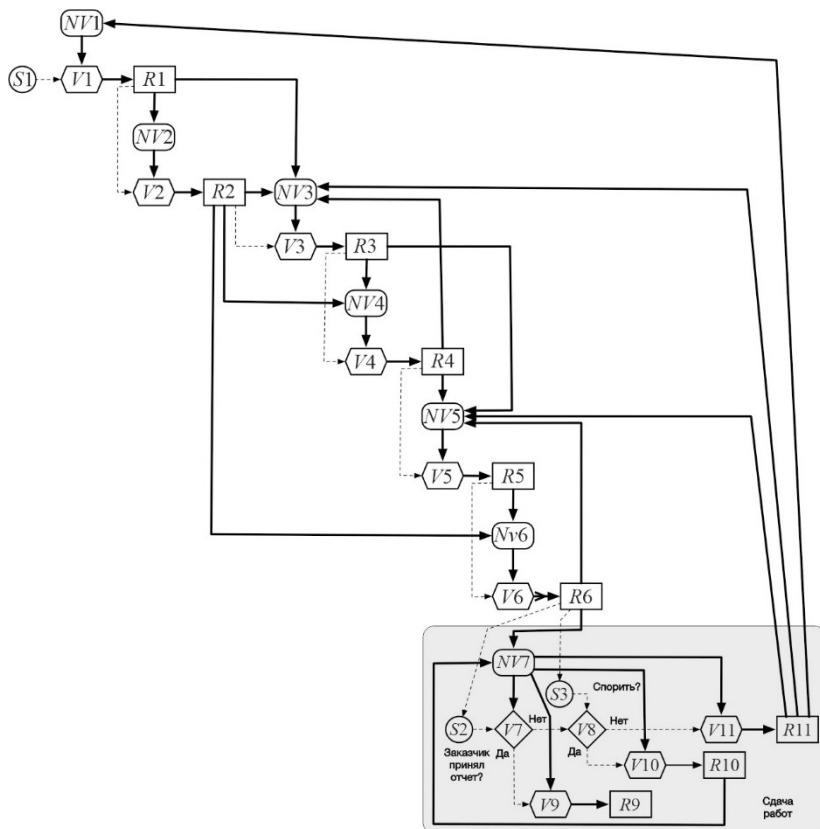


Рис. 36. Структурно-функциональная схема организации процесса подготовки строительной документации до изменений (соответствие обозначений операциям см. в табл. 10)

Для представления структурно-функциональной модели после изменений введем обозначения, представленные в табл. 3, и получим схему, приведенную на рис. 37.

Исследовать полученные сети Петри и структурно-функциональные модели будем методами статистического моделирования, для чего проведем серию экспериментов.

Таблица 11

Система обозначений для структурно-функциональной модели процесса подготовки строительной документации после внесения изменений и выявленные соответствия с приведенной на рис. 2 сетью Петри

Обозначение	Соответствия на сети Петри (см. рис 2)	Время обработки	Сложность	Описание
$S1$	$P_{нач}$	–	–	Поступление нового проекта
$V1$	$P_{нач}$	2	k	Вынесение решения о принятии проекта в работу
$NV1$	$P_{нач}$	–	–	Данные о свободных ресурсах
$R1$	$P_{нач}$	–	–	Множество проектов, принятых в работу
$NV2$	A	–	–	Техническое задание проекта
$V2$	A, Z_1, Z_2	18	k	Сбор информации / исходных данных
$R2$	A	–	–	Исходные данные проекта
$NV3$	A	–	–	Исходные данные проекта
$V3$	B	8	k	Формирование концепции проекта
$R3$	B	–	–	Пояснительная записка
$NV4$	C	–	–	Концепция проекта
$S2$	$C, U1$	–	–	Реакция заказчика на материалы 1-го этапа
$S3$	$U1$	–	–	Реакция исполнителя на замечания к концепции
$V6$	$O1$	4	k	Подготовка ответа на замечания
$R6$	$O1$	–	–	Ответы на замечания
$V7$	$p2$	2	k	Обработка / учёт замечаний
$R7$	$p2$	–	–	Материалы для доработки
$V8$	$p2, D$	40	k	Оформление документации
$R8$	D	–	–	Комплект документов
$NV9$	D	–	–	Документация
$S4$	$E, U2$	–	–	Реакция заказчика на материалы 2-го этапа
$S5$	$U2$	–	–	Реакция исполнителя на замечания к документации
$V12$	$O2$	4	k	Подготовка ответа на замечания

Обозначение	Соответствия на сети Петри (см. рис 2)	Время обработки	Сложность	Описание
R_{12}	O_2	—	—	Ответы на замечания
V_{13}	p_3	2	k	Обработка/учёт замечаний
R_{13}	p_3	—	—	Материалы для доработки
V_{11}	$p_{кон}$	4	k	Передача материалов заказчику, подписание актов
R_{11}	$p_{кон}$	—	—	Разработанные и сданные заказчику документы

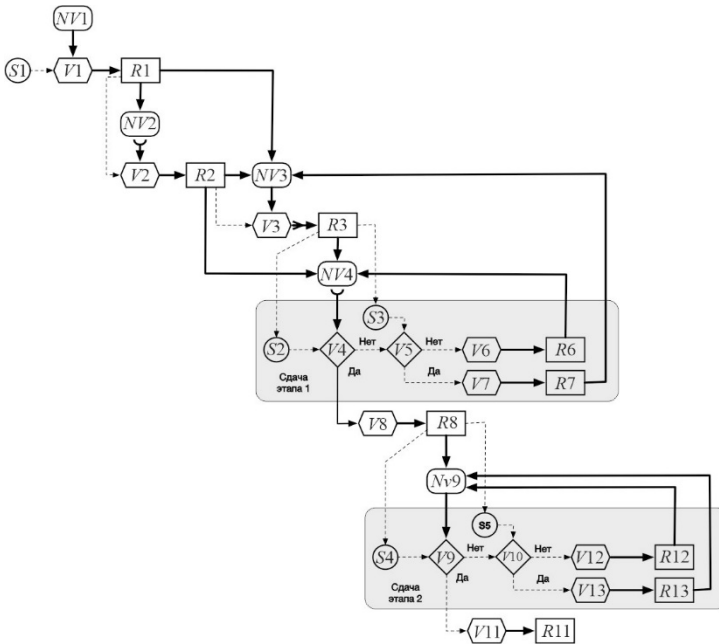


Рис. 37. Структурно-функциональная модель организации процесса подготовки строительной документации после изменений (соответствие обозначений операциям см. в табл. 11)

Эксперимент 1. Исследование времени обработки запросов (поступающих проектов).

Для получения статистических данных будем производить моделирование на сетях Петри. В местах разветвлений на моделях (см. рис. 1, 2) указаны значения, которые показывают, с какой вероятностью пойдет процесс по каждому из путей. Эти значения были получены экспертным методом. Имея эти данные, промоделируем 100 раз (т.е. на примере 100 заказов) процесс разработки строительной документации. Для проведения экспериментов была написана программа на python, моделирующая поток заявок по сетям Петри. Результатом выполнения программы является массив уникальных идентификаторов мест, через которые прошла фишка (табл. 12).

Таблица 12

Примеры данных проведения заказов по подготовке строительной документации на основе сетей Петри, приведенных на рис. 33, 34

Эксперименты в сети «как есть»			
1	2	3	...
$P_{нач}$	$P_{нач}$	$P_{нач}$...
A	A	A	
Z1	B	B	
V	C	C	
Z2	U1	U1	
A	A	A	
B	B	B	
C	C	C	
D	U1	D	
F	A	F	
E	B	U2	
$P_{кон}$	C	D	
	U1	F	
	A	U2	
	B	D	
	C	F	
	D	E	
	F	$P_{кон}$	
	E		
	U3		
	D		
	F		
	U2		
	D		
	F		
	E		
	$P_{кон}$		

Эксперименты в сети «как должно быть»				
1	2	3	4	...
$P_{нач}$	$P_{нач}$	$P_{нач}$	$P_{нач}$...
A	A	A	A	
B	Z1	B	Z1	
C	Z2	C	Z2	
U1	A	D	A	
O1	Z1	E	Z1	
C	Z2	$p0$	Z2	
D	A		A	
E	Z1		B	
U2	Z2		C	
O2	A		U1	
E	B		A	
$P_{кон}$	C		B	
	D		C	
	E		U1	
	U2		O1	
	O2		C	
	E		D	
	$P_{кон}$		E	
			U2	
			O2	
			E	
			$P_{кон}$	

Зная время отдельных операций (см. табл. 12) и просуммировав времена выполнения для каждого эксперимента, получаем общее время, затраченное на проект, выполняемый в моделируемой системе (табл. 13).

Таблица 13

Примеры результатов времени реализации проектов, полученные на основе моделирования

Параметр	1	2	3	4	...
До изменений структуры процесса	108	280	670	786	...
После изменения структуры процесса	116	148	80	186	...

Проанализировав данные 100 экспериментов на каждой из моделей, получим:

- **средние значения:** до изменений структуры процесса – 326,52 ч, после изменений структуры процесса – 180,88 ч;
- **медианное значение:** до изменений структуры процесса – 248 ч, после изменений структуры процесса – 152 ч.

Построим скрипичные диаграммы (рис. 38).

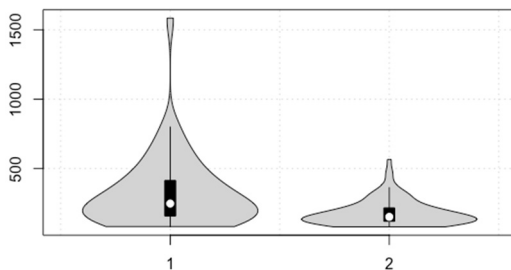


Рис. 38. Скрипичные диаграммы: 1 – до изменений структуры процесса, 2 – после изменения структуры процесса

В результате можно заметить, что распределение значений времени выполнения процесса и отклонение от него сокращается, что говорит о более эффективной и более стабильной организации процесса и подтверждается значениями коэффициента Джини

(до изменений структуры процесса – 0,3786928825186819, после изменения структуры процесса – 0,264734630694383) [71].

Эксперимент 2. Сравнение моделей по величине цикломатической сложности.

Так как мы имеем последовательность действий, формализованную в виде графа (сеть Петри), то мы определим количество рёбер в графе (E), количество узлов в графе (N), количество компонент связности (P) и на их основе посчитаем цикломатическую сложность по формуле для сильносвязанных графов $M (M = E - N + P)$ (табл. 14), которая покажет, какой способ организации процесса менее склонен к ошибкам реализации [14] и будет проще в организации.

Таблица 14

Оценки цикломатической сложности на основе сетей Петри

Параметр	E	N	P	M
До изменений структуры процесса	42	36	1	$42-36+1=7$
После изменения структуры процесса	30	26	1	$30-26+1=5$

Аналогичные оценки можно провести и для графов, формализованных в форме моделей с выделенным субъектом управления (табл. 15).

Таблица 15

Оценки цикломатической сложности на основе структурно-функциональных моделей с выделенным субъектом

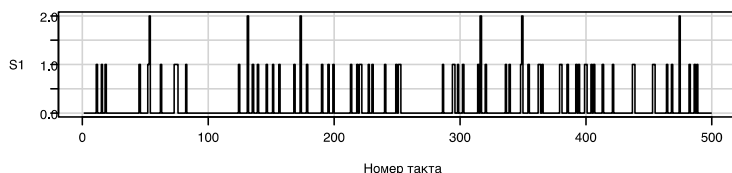
Параметр	E	N	P	M
До изменений структуры процесса	41	28	1	$41-28+1=14$
После изменения структуры процесса	37	28	1	$37-28+1=10$

Из приведенных оценок видно, что для сети Петри цикломатическая сложность меньше для нового варианта; то же самое справедливо и для структурно-функциональных моделей. Таким образом, можно говорить, что новый вариант является менее подверженным к ошибкам, возникающим в процессе создания и работы информационной инфраструктуры.

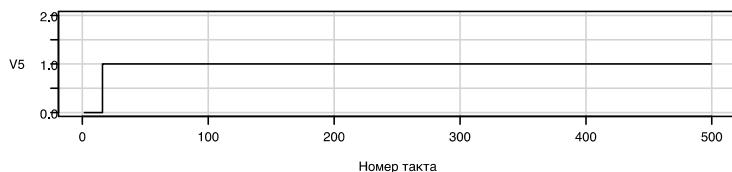
Приведенное сравнение протекания процессов рассматривает процесс в условиях равномерного поступления заявок, поэтому необходимо дополнительное исследование на устойчивость к нагрузкам. Для этого будем использовать модель с выделенным субъектом управления.

Эксперимент 3. Исследование поведения моделей при неравномерном поступлении заявок.

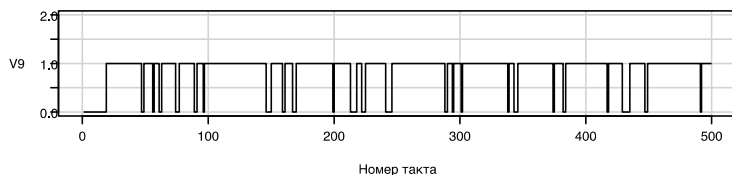
Для проведения численных расчетов на структурно-функциональной модели необходима информация о том, как будет осуществляться управление.



a



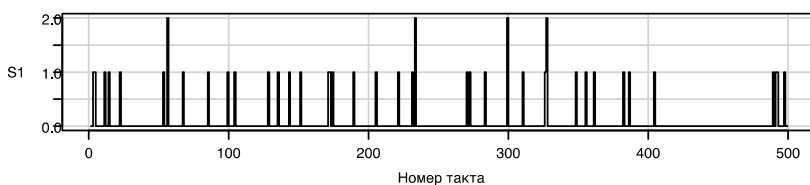
б



в

Рис. 39. Периодограммы для модели до изменений: *a* – поступления проектов; *б* – выполнения операции оформления документов; *в* – выхода готовых проектов

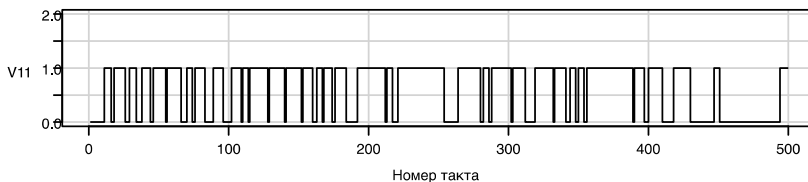
Случайную величину представляет собой время появления нового проекта, что может быть промоделировано вероятностью его появления ($S1$); аналогично будем моделировать оспаривание результатов сдачи и отправку на доработку ($S2$). Если принять один такт расчета за два часа и провести моделирование в течении 500 тактов, тогда для структурно-функциональной модели до изменений получим результаты по загрузке операционных элементов, которые приведены на рис. 39, а после предлагаемых изменений – результаты, приведенные на рис. 40.



a



б



в

Рис. 40. Периодограммы для модели после изменений: *a* – поступления проектов; *б* – выполнения операции оформления документов; *в* – выхода готовых проектов

Из приведенных рисунков видно, что снижается загрузка основного операционного элемента, связанного непосредственно с подготовкой текстов документов, а также снижение частоты появления проектов на выходе, что косвенно говорит о том, что проекты до приемки заказчиком проходят меньшее число возвратов.

Используемая нотация позволяет проводить исследование длительности реализации проектов в тактах и загрузку операционных элементов (рис. 41).

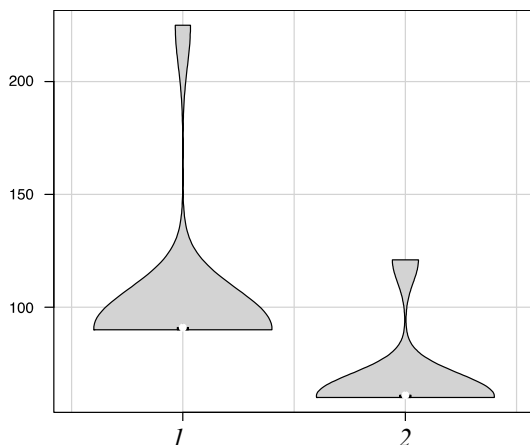


Рис. 41. Скрипичные диаграммы времени выполнения проектов: 1 – до изменений структуры процесса, 2 – после изменения структуры процесса

Из полученных графиков видно, что происходит общее снижение загрузки операционных элементов, задействованных в реализации проектов, а также снижение общего времени реализации проекта и длины «хвоста» для проектов, возвращаемых на доработку.

Таким образом, в результате использования разных подходов и нотаций, несмотря на их отличия, мы получаем сходные результаты, которые подтверждают лучшие характеристики новой предлагаемой структуры организации проекта.

Кроме приведенных выше, могут использоваться и другие показатели для выбора способа организации процесса. Такие, напри-

мер, как затраты на перевод системы в новое состояние, затраты на обеспечение функционирования процесса, потребности в вычислительных мощностях и др. Тогда задача выбора будет сводиться к задаче многокритериального ранжирования и потребует использования специальных методов [17] (метод из группы методов экспертного выбора или таких, как TOPSIS, ELECTRE, VIKOR).

Кроме этого, стоит отметить, что качество исследования будет зависеть от того, насколько хорошо мы генерируем последовательности случайных событий и насколько верно мы выбрали закон распределения для их описания.

Глава 3. ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ ДЛЯ ИССЛЕДОВАНИЯ ПРОЦЕССОВ МЕТОДАМИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ*

Генерация случайных чисел слишком важна, чтобы оставлять её на волю случая.

Роберт Кавью

Для использования генератора случайных чисел необходимо знать его основные характеристики:

- 1) способ запуска;
- 2) размер цикла;
- 3) время генерации случайного числа i ;
- 4) статистические характеристики.

Говоря о последовательностях независимых случайных чисел с определенным законом распределения, мы имеем в виду, что каждое число получено произвольным образом, без какой-либо связи с другими членами последовательности и что оно имеет определенную вероятность нахождения в произвольно заданном интервале.

В равномерном распределении, где все возможные числа равновероятны, каждая из десяти цифр от 0 до 9 представляет собой одну десятую часть всех цифр в каждой случайной последовательности цифр. Каждая заданная пара двух соседних цифр должна представлять 1/100 всех пар, встречающихся в последовательности, и т.д. Однако, если мы посмотрим на конкретную случайную последовательность из миллиона цифр, она, конечно, не обязательно должна содержать ровно 100 000 нулей, 100 000 единиц и т.д. На самом деле вероятность такого события очень мала. Регулярность выполняется в среднем для последовательностей таких последовательностей.

Произвольно заданная последовательность столь же вероятна, как и последовательность, состоящая только из нулей. Кроме того,

* Глава написана З. Аврамовичем, Белградский университет.

предположим, что мы случайным образом выбираем последовательность из одного миллиона цифр. Пусть случится так, что первые 999 999 цифр – все нули. Даже в этом случае вероятность того, что последняя цифра будет равна нулю, по-прежнему равна 1/10, если выбор действительно случайный.

Кажется, что эти утверждения звучат парадоксально, но противоречия в них нет.

Используется несколько общепринятых процедур генерации псевдослучайных чисел. Это все итерационные процедуры, где псевдослучайное число получается из предыдущего или нескольких предыдущих:

$$X_i = f(X_{i-1})$$

или

$$X_i = f(X_{i-1}, X_{i-2}, \dots).$$

Процедура генерации псевдослучайных чисел начинается с определения первого случайного числа X_0 , называемого начальным числом. Запуском программы в компьютере определяется X_0 (как правило, это число задается), а затем для получения случайного числа обычно вызывается подпрограмма, работающая с параметрами SEME (входной параметр) и X_i (выходной параметр). Входным параметром является начальное число, которое позволит получить значение выходного параметра X_i с помощью запрограммированной процедуры. Например, в FORTRAN CALL GSB(SEME, X_i) означает, что $X_i = f(\text{SEME})$. После выхода из подпрограммы необходимо переопределить начальное число в виде $\text{SEME} = X_i$, что подготавливает повторное использование подпрограммы, то есть генерацию нового, очередного псевдослучайного числа в последовательности.

Многие генераторы случайных чисел, популярные в настоящее время, недостаточно хороши. Нередко бывает так, что старый, сравнительно несовершенный метод передается от одного разработчика к другому вслепую, и современный пользователь ничего не знает о его недостатках. Однако изучить наиболее важные свойства генераторов случайных чисел и научиться применять эти знания несложно.

3.1. Способы получения случайных чисел

Ранее исследователи, которым для работы были нужны случайные числа, использовали карты, бросали игральные кости или вытягивали шары из урны, которую предварительно как следует встряхивали. Сегодня для этой цели используют в основном три метода:

- таблицы случайных чисел;
- генерацию случайных чисел электронными элементами (шумы в электронных устройствах);
- методы генерации псевдослучайных чисел.

Использование таблиц случайных чисел

В 1927 году Л.Г.К. Типпетт напечатал таблицы, содержащие более 40 000 случайных цифр, «произвольно взятых из переписного отчета». Позже были сконструированы специальные машины, которые механически «производили» случайные числа. Первая такая машина была использована в 1939 году М.Дж. Кендаллом и Б. Барбингтоном-Смитом при создании таблиц со 100 000 случайных цифр. В 1955 году корпорация RAND* опубликовала знаменитые таблицы миллиона случайных чисел, полученных с помощью специально сконструированного колеса рулетки. Хотя эта процедура обладает всеми свойствами случайности, через длительный промежуток времени были получены цифры, которые, согласно статистической проверке, уже не имели равномерного распределения.

С помощью таблиц можно формировать случайные числа, выбирая цифры, начиная с любого места таблицы в произвольном направлении. Таким образом, получаем двузначные, трехзначные или многозначные случайные числа, а также числа, распределенные между 0 и 1.

Методы генерации псевдослучайных чисел

С появлением компьютеров началась и разработка методов получения случайных чисел, пригодных для использования в программных системах. Можно применять таблицы, хотя этот подход связан с использованием большого числа памяти и временем на

ввод данных в ЭВМ, требует затрат и создания самих таблиц. Противоположностью этому является высокая скорость вычислительных процедур, которые хотя и не дают случайных чисел, но представляют собой удовлетворительную замену им.

Псевдослучайные числа имеют преимущества перед чистыми случайными числами, связанные с тем, что:

1. Компьютер может генерировать их по какой-то проверенной процедуре (формуле) генерации, поэтому не требуется места в памяти компьютера для размещения больших таблиц (заранее подготовленных случайных чисел) или машинного времени, которое тратится на их загрузку, если они хранились на периферийных устройствах.

2. Пользователю не нужно все время использовать один и тот же источник случайных чисел. Путем изменения параметров, характерных для применяемого алгоритма, получается новая последовательность чисел.

3. Появляется возможность воспроизведения серии псевдослучайных чисел, которая статистически очень важна, особенно на этапе тестирования программ.

Поскольку качество случайных чисел определяется не процессом их получения, а тем, удовлетворяют ли они некоторым критериям случайности, то не имеет значения, какой процесс мы использовали для их получения.

Метод средних квадрата

Первая процедура получения случайных чисел с помощью компьютерных арифметических операций была предложена в 1946 году Джоном фон Нейманом. Он назвал её методом средних квадрата. Идея состоит в том, чтобы возвести в квадрат предыдущее «-значное» случайное число, а затем извлечь из результата j средних цифр. Например, рассмотрим четырехзначное число и предположим, что начальное число

$$X_0 = 5139,$$

возведя в квадрат, $X_0^2 = 26(4093)21$, поэтому следующее число $X_1 = 4093$,

возведя в квадрат, $X_1^2 = 16(7526)49$, поэтому следующее число будет $X_2 = 7526$, и т.д.

Аргумент против метода состоит в том, что последовательность, полученная таким образом, не может быть случайной, если каждое новое значение полностью определяется предыдущим. Однако получаемая последовательность, хотя и не случайная, кажется случайной. В типичных приложениях обычно не имеет значения, как связаны друг с другом два соседних числа. Такой неслучайный характер последовательности не является нежелательным. Интуитивно понятно, что среднеквадратический метод должен достаточно хорошо «изменять» предыдущее число.

На практике метод среднего квадрата Неймана оказался плохим источником случайных чисел. Его недостаток состоит в том, что строки имеют тенденцию превращаться в короткие циклы повторяющихся элементов. Например, если один член массива станет равным нулю, все последующие члены также будут равны нулю. В начале 50-х годов несколько ученых проанализировали метод средних квадратов. Э. Форсит, работая с четырехзначными числами, проверил 16 чисел в роли начальных значений строк. Оказалось, что 12 из 16 заканчивались циклом 6100, 2100, 4100, 8100, 6100, ... и еще 2 выродились в нули. Множественные эксперименты по исследованию метода среднего квадрата были выполнены Н.К. Метрополисом, который работал в основном с двоичными числами. Работая с двадцатизначными числами, он показал, что существует 13 различных циклов, в которые можно преобразовать строки. Длина периода наибольшего из них равна 142. Когда строка обращается в ноль, довольно легко снова начать генерировать случайные числа, дополнив программу специальной процедурой для установки нового начального числа. На основе этого Р. Флойд предложил метод, позволяющий регистрировать появление циклов в последовательности. Метод Флойда требует небольшой загрузки памяти компьютера, увеличивает время генерации случайных чисел всего в три раза и фиксирует появление в последовательности ранее сгенерированного числа.

Алгоритм Флойда:

Шаг 1: устанавливаем $X_0 \rightarrow X \rightarrow Y$.

Шаг 2: выводим X .

Шаг 3: выполняем $f(X) \rightarrow Y$

$$f(f(Y)) \rightarrow H.$$

Шаг 4: если $H \neq Y$, то переходим к шагу 2.

Н.К. Метрополис, работая с 38-значными двоичными числами, нашел последовательность, состоящую из 750 000 взаимно различных членов. Статистические тесты подтвердили случайный характер полученной последовательности размером $750\,000 \times 38$ бит. Это подтвердило, что можно получить полезные результаты, применяя метод среднего квадрата. Из-за высокой зависимости (с точки зрения длины цикла и возможности вырождения) от выбора начального числа, метод не является слишком надежным. Кроме этого, такой генератор еще и медленный (по сравнению с другими методами), поэтому используется довольно редко.

Метод средних результата

Идея, которая лежит в основе данного метода, очень похожа на используемую в предыдущем методе. Следующее число получается путем умножения предыдущего числа на некоторую константу K и извлечения средних цифр произведения. Например, для генерации двузначных случайных чисел с выбором константы $K = 41$ и начальным числом $X_0 = 29$ мы получаем следующую последовательность:

$$\begin{aligned} X_0 &= 29, \\ K \cdot X_0 &= 41 \cdot 29 = 1(18)9, \text{ поэтому } X_1 = 18, \\ K \cdot X_1 &= 41 \cdot 18 = 0(73)8, \text{ поэтому } X_2 = 73, \end{aligned}$$

и т.д.

Недостатком этого алгоритма является то, что часто происходит вырождение (например, $K = 50$ и $X_0 = 38$ дают последовательность: 90, 50, 50, ..., 50). Но если сделать значение K переменным,

формируя следующее случайное число из средних цифр произведения двух предыдущих случайных средних чисел, такого рода вырождения можно избежать. Следовательно, вначале необходимо иметь два «семени» X_0 и X_1 . Предположим, что это

$$X_0 = 38, \text{ а } X_1 = 51,$$

$$\text{тогда } X_0 \cdot X_1 = 1(93)8, \text{ отсюда } X_2 = 93,$$

$$\text{далее } X_1 \cdot X_2 = 4(74)3, \text{ поэтому } X_3 = 74,$$

и т.д.

Получаем последовательность: 38, 51, 93, 74, 88, 51, 48, 44, 11, Несмотря на то, что второй и шестой члены последовательности одинаковы, цикл не возник.

Использовать этот генератор, несмотря на то, что он лучше, чем генератор средних квадрата, согласно тестам на однородность тоже не рекомендуется.

Создать простой генератор случайных чисел не так просто. Над этой задачей работал и такой известный в области информатики учёный, как Д. Кнут.

3.2. Равномерно распределенные случайные числа

Для получения последовательностей случайных вещественных чисел U_n , равномерно распределенных между нулем и единицей, будем генерировать целые числа X_n в интервале от 0 до некоторого m , так как в компьютере вещественное всегда представляется с ограниченной точностью. Тогда значение

$$U_n = \frac{X_n}{m}$$

попадает в интервал от нуля до единицы. Обычно m на единицу больше, чем максимальное число, которое может поместиться в машинном (компьютерном) слове.

Линейно-конгруэнтный метод (генератор случайных чисел Лемера)

Лучшие из известных сегодня генераторов случайных чисел представляют собой частные случаи следующей схемы, предложенной Д. Лемером в 1948 году. Выберем четыре «магических» числа.

X_0 , начальное значение, ($X_0 \geq 0$);

a , множитель, ($a \geq 0$);

s , приращение, ($s \geq 0$);

m , модуль, ($m \geq 1$, $m > a$, $m > s$).

Тогда искомая последовательность случайных чисел $\{X_n\}$ получается из соотношения

$$X_{n+1} = (a \cdot X_n + s) \bmod m, \quad n \geq 0.$$

Она называется линейной конгруэнтной последовательностью.

Если $a = 1$, метод называется аддитивно-конгруэнтным методом, а термины «мультипликативный конгруэнтный» и «смешанный конгруэнтный» обозначают линейные конгруэнтные методы при $s = 0$ и $a \neq 1$ и $s \neq 0$ соответственно.

Например, чтобы сгенерировать числа от 0 до 9 ($m = 10$) и выбрав $X_0 = 7$, $a = 3$, $s = 4$, мы составляем следующую последовательность:

$$X_0 = 7,$$

$$a \cdot X_0 + s = 3 \cdot 7 + 4 = 25, \quad 25 \bmod 10 = 5, \quad \text{поэтому } X_1 = 5,$$

$$a \cdot X_1 + s = 3 \cdot 5 + 4 = 19, \quad 19 \bmod 10 = 9, \quad \text{поэтому } X_2 = 9,$$

и т.д.,

то есть (7, 5, 9, 1, 7, 5, 9, 1, ...). Цикл повторяется после четвертого члена.

Увеличение константы m обычно дает более длинную последовательность, но неправильный выбор констант может привести

к вырожденной последовательности. Например, при $m = 80$ и различных значениях констант a , s и X_0 мы получаем следующие последовательности:

$a = 29, s = 10, X_0 = 31, (31, 29, 51, 49, 71, 69, 11, 9, 31, \dots)$ – повторение цикла,

$a = 15, s = 10, X_0 = 31, (31, 75, 15, 75, 15, \dots)$ – цикл с двумя членами,

$a = 50, s = 0, X_0 = 10, (10, 20, 40, 0, 0, \dots)$ – вырождение.

Как хорошо видно, последовательность не всегда будет казаться «случайной», если X_0, a, s, m выбраны произвольно.

Приведенный пример иллюстрирует тот факт, что конгруэнтные последовательности сводятся к циклам, т.е. в конце концов числа образуют цикл, который повторяется бесконечное количество раз. Этим свойством обладают все последовательности общего вида $X_{n+1} = f(X_n)$. Повторяющийся цикл называется периодом. Длина периода в рассмотренных последовательностях невелика. Реально используемые последовательности имеют, конечно, несравненно больший период.

Частный случай $s = 0$ приводит к ускорению получения случайных чисел. Однако тогда уменьшается длина периода последовательности, но даже в этом случае можно получить большой период. В первом предложенном методе Лемера использовалось $s = 0$, хотя автор упоминал возможность использования $s \neq 0$. Идея получения более длинных последовательностей за счет обобщения $s \neq 0$ принадлежит С. Томсону и А. Ротенбергу.

Для упрощения некоторых формул полезно определять b как

$$b = a - 1.$$

Случай $a = 1$ можно сразу отбросить, так как тогда $X_n = (X_0 + n \cdot s) \bmod m$, и, очевидно, что последовательность неслучайна. Вариант $a = 0$ еще хуже. Следовательно, для практических приложений можно считать, что

$$a \geq 2, b \geq 1.$$

В результате можно обобщить исходное соотношение,

$$X_{n+k} = (a^k \cdot X_n + (a^k - 1) \cdot \frac{S}{b}) \bmod m, \quad k \geq 0, \quad n \geq 0$$

выражая $(n + k)$ -й член через k (следует обратить внимание на частный случай $n = 0$). Последовательность, состоящая из каждого «-го» члена нашей последовательности, образует второй линейный конгруэнтный метод с множителем a_k и приращением $\left((a^k - 1) \cdot \frac{S}{b} \right)$.

Группа случайных чисел (периодов) должна быть достаточно большой для получения большого числа «случайных» чисел. Для этого нам следует сделать значение m как можно большим, поскольку длина периода не может быть больше m .

Другим фактором, влияющим на выбор m , является скорость генерации чисел: мы хотим выбрать такое значение, чтобы как можно быстрее вычислить $(a \cdot X^n + s) \bmod m$.

Рассмотрим, как правильно выбрать значение m

Период получаемых случайных чисел должен быть достаточно большой. Для этого мы стараемся сделать значение m как можно большим, потому что длина периода не может быть больше m . Еще одним фактором, влияющим на выбор m , является скорость генерации чисел. Желательно выбрать такое значение, чтобы вычислять $(aX_n + c) \bmod m$ как можно быстрее.

Предположим, что $Z = (aX_n + c)$ и m – положительные числа, и если мы хотим вычислить $Z \bmod m$, то легко увидеть, что, разделив на m , мы можем получить желаемый результат. Но поскольку деление довольно медленная операция, мы можем избежать ее, присвоив m значение размера слова (т.е. большее, чем максимальное целое число, которое может уместиться в машинном слове).

Пусть n_j – это максимальное целое число. Тогда операция сложения выполняется по модулю n_j , и умножение по модулю n_j относительно просто.

Известны и быстрые методы выполнения операций умножения и сложения по модулю для $(n_j + 1)$. Вполне уместным может быть вопрос: как можно записать в регистр столько значений? Очевидно, что числа больше $(n_j - 1)$ не могут быть помещены в регистр. Ответ заключается в том, что переполнение происходит тогда и только тогда, когда результат равен n_j .

Обоснован также вопрос, почему $m = n_j$ менее выгодно, чем $m = n_j + 1$. Опыт показывает, что при $m = n_j$ младшие разряды числа X_n менее случайны, чем старшие. При выборе $m = n_j + 1$ младшие цифры ведут себя так же случайно, как и старшие.

Одна из возможностей – выбрать значение m как наибольшее простое число, меньшее n_j .

Для большинства приложений младшие разряды не имеют значения, и $m = n_j$ является удовлетворительным выбором, если программист, использующий случайные числа, делает это разумно.

Рассмотрим, как выбрать множитель a , чтобы получить период максимальной длины

Для произвольной последовательности, предназначенной для использования в качестве источника случайных чисел, важен максимально длинный период. Обычно нам хотелось бы, чтобы период содержал значительно больше чисел, чем необходимо для решения конкретной задачи.

Большой период – лишь одно из свойств случайных последовательностей. Возможны абсолютно неслучайные последовательности с очень большим периодом. Например, при $a = s = 1$ последовательность просто сводится к $X_{n+1} = (X_n + 1) \bmod m$. Очевидно, её период равен m , но случайной такую последовательность назвать нельзя.

Поскольку можно получить только m разных значений, длина периода не может быть больше m . Достижима ли максимальная длина m ? Приведенный выше пример показывает, что это так, хотя выбор $a = s = 1$ не приводит к желаемой последовательности.

Выразим возможные способы выбора a и s , которые дают период длины m . (Примечание: поскольку длина периода равна m , каждое число от 0 до $(m - 1)$ встречается за период только один раз. В этом случае выбор X_0 не влияет на длину периода.)

Теорема 3.1. *Длина периода линейной конгруэнтной последовательности равна m тогда и только тогда, когда:*

- 1) s и m – взаимно простые числа;
- 2) $b = a - 1$, кратно r , для произвольного простого числа r , являющимся множителем числа m ;
- 3) b делится на 4, если m делится на 4.

Идея доказательства известна уже более ста лет. Первое доказательство сформулированной теоремы в таком виде было дано М. Гринбергером для частного случая $m = 2^e$. Т. Хул и А. Добелл доказали, что условия 1), 2) и 3) достаточны в общем случае (см. [72]).

Рассмотрим частный случай чисто мультипликативного генератора, который получаем, если принять $s = 0$. Хотя в этом случае генерация случайных чисел происходит несколько быстрее, из теоремы 3.1 следует, что максимальную длину периода получить невозможно. Естественно, это вполне ожидаемо, поскольку члены последовательности удовлетворяют соотношению

$$X_{n+1} = a \cdot X_n \pmod{m},$$

и значение $X_n = 0$ можно найти в нем только в том случае, если последовательность сводится к нулю. В общем, если d – произвольный делитель m и если X_n делится на d , все последующие значения X_{n+1} , X_{n+2} , ... также делятся на d . Следовательно, при $s = 0$ желательно, чтобы X_n и m были взаимно простыми для каждого n , и это ограничивает длину периода.

Получить приемлемо длинный период можно и при $s = 0$, если, конечно, стремиться к этому. Теперь попробуем найти такие условия, по которым определяется множитель, чтобы длина периода была максимальной даже в этом частном случае.

Обозначим через $g(m)$ максимально возможный порядок по модулю m . Не представляет особой сложности дать точные значения $g(m)$ в следующих случаях:

$$g(2) = 1, \quad g(4) = 2, \quad g(2^e) = 2^{e-2}, \quad \text{если } e \geq 3,$$

$$g(p^e) = p^{e-1} \cdot (p-1), \quad \text{если } p > 2,$$

$$g(p^{e_1} \cdot \dots \cdot p^{e_r}) = NZS(g(p^{e_1}), \dots, g(p^{e_r})).$$

На основе описанных наблюдений можно сформулировать следующую теорему.

Теорема 3.2 (теорема Р. Кармайкла [73]). *Максимально возможный период при $s = 0$ равен $g(m)$, где $g(m)$ определяется выражениями, приведенными выше. Этот период реализуется, если выполняется*

- 1) X_0 и m – взаимно простые числа;
- 2) a – прототип переменной $\text{mod } m$.

Если m – простое число, то можно получить период длиной $m - 1$, что всего на 1 меньше максимально возможного. Вопрос в том, как найти a , которое позволит этого достичь.

Теорема 3.3. *Число a является прототипом $\text{mod } p^e$ тогда и только тогда, когда:*

1) $p^e = 2$, a – нечетное; или $p^e = 4$, и $a \text{ mod } 4 = 3$; или $p^e = 8$, и $a \text{ mod } 8 = 3, 5, 7$; или $p = 2$, $e \geq 4$ и $a \text{ mod } 8 = 3$ или 5 ;

2) p – нечетное, $e = 1$, $a \neq 0$ (по $\text{mod } p$) и $a^{(p-1)j} \neq 1$ (по $\text{mod } p$) для произвольного простого делителя j числа $p - 1$;

3) p – нечетное, $e > 1$, удовлетворяет условию 2) и $a^{p-1} \neq 1$ (по $\text{mod } p^2$).

Для важного случая $m = 2^e$ при $e \geq 4$ приведенные выше условия сводятся к единственному требованию, чтобы $a \equiv 3$ или 5 (по $\text{mod } 8$). В этом случае четвертая часть всех возможных множителей дает максимальный период.

Второй наиболее распространенный случай – $m = 10^e$. Следующая теорема определяет необходимые и достаточные условия для достижения максимального периода.

Теорема 3.4. *Если $m = 10^e$, $e \geq 5$, $s = 0$ и X_0 не делится на 2 или 5, период линейной конгруэнтной последовательности равен $5 \cdot 10^{e-2}$ тогда и только тогда, когда $\text{mod } 200$ принимает одно из следующих 64 значений – $\pm \{3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91\}$.*

Рассмотрим, как определить степень генератора случайных чисел

Мы только что показали, что можно получить максимальный период при $b = a - 1$, делящийся на все простые делители числа m (b должен делиться на 4, если m делится на 4). Если z – основание системы исчисления, которое используется в компьютере (например, $z = 2$ для двоичных компьютеров и $z = 10$ для десятичных компьютеров), m – размер машинного слова ($m = ze$), то множитель

$$a = z^k + 1, 2 \leq k < e$$

соответствует всем условиям. Из теоремы 3.1 следует, что можно взять $s = 1$. Тогда рекуррентное соотношение примет форму

$$X_{n+1} = ((z^k + 1) \cdot X_n + 1) \text{ mod } z^e.$$

Во время вычислений мы можем избежать умножения за счет простого сложения и сдвига.

Множители вида $a = z^k + 1, 2 \leq k < e$ подробно обсуждались в литературе и рекомендовались многими авторами. Однако эмпирические исследования показывают, что таких простых множителей лучше избегать в связи с тем, что на многих современных компьютерах умножение выполняется быстрее, чем сдвиг и сложение! Однако самый серьезный аргумент против использования множителя вида $z^k + 1$ состоит в том, что он дает недостаточно случайные числа. Одна из причин этого связана с понятием «стадия генератора», которую мы сейчас и рассмотрим.

Степень линейной конгруэнтной последовательности с максимальным периодом определяется как наименьшее целое число s такое, что

$$b^s \equiv 0 \pmod{m}.$$

Такое целое число s существует всегда, если множитель удовлетворяет условиям теоремы 3.1, в частности, если b делится на произвольный простой множитель m .

Можно проанализировать случайность последовательности, приняв $X_0 = 0$, поскольку в течение периода обязательно выпадет ноль. В этом случае мы имеем $X_n = \left((a^n - 1) \cdot \frac{c}{b} \right) \pmod{m}$ и, разделив $a^n - 1 = (b + 1)^n - 1$ по биномиальной формуле, получим:

$$X_n = c \left(n + \binom{n}{2} b + \dots + \binom{n}{s} b^{s-1} \right) \pmod{m}.$$

Все члены b^s, b^{s+1} и т.д. мы можем опустить, поскольку они делятся на m .

Исходя из последнего выражения, рассмотрим несколько частных случаев:

а) если $a = 1$, то степень равна 1 и, как мы знаем, $X_n \equiv cn \pmod{m}$, поэтому последовательность явно неслучайна;

б) если степень равна 2, последовательность нельзя считать случайной. Зависимость между соседними элементами представляет собой простую функцию от n : $X_{n+1} - X_n \equiv c + cbn$;

в) если степень равна 3, последовательность выглядит более случайной. Тем не менее X_n, X_{n+1} и X_{n+2} сильно зависят друг от друга;

г) казалось, что приемлемые результаты можно получить и со степенью 4, но многие авторы это оспаривают;

д) достаточно случайные величины можно получить со степенью не ниже 5.

Пример. Предположим, $m = 2^{35}$ и $a = 2^k + 1$. Тогда $b = 2^k$, поэтому $k \geq 18$ значение $b^2 = 2^{2k}$ делится на m , степень равна 2. Если $k = 17, 16, \dots, 12$ степень равна 3, а при $k = 11, 10, 9$ достигает значения 4. Следовательно, с точки зрения степени, приемлемыми являются такие множители, для которых $k \leq 8$. Это означает, что $a \leq 257$. Позже мы увидим, что следует избегать и малых множителей. Таким образом, все множители вида $2^k + 1$ при $m = 1^{35}$ неприемлемы.

Для более длинных слов можно использовать множители типа $2^k + 1$. В литературе известен и описан генератор с $m = 2^{47}$, $a = 2^9 + 1$ и степени, равной 6. Генератор с такими параметрами не проходит даже статистических проверок.

Если m равно $n_j + 1$, где n_j – размер слова, и m не разлагается на произведения простых чисел высокой степени, поскольку генератор высокой степени невозможен. Следовательно, в этом случае не следует использовать метод максимального периода, а следует использовать метод чистого умножения $s = 0$.

В любом случае в выборе множителя по-прежнему остается большая свобода. В целом мы хотим сохранить высокую степень генератора случайных чисел, достаточно большой множитель и в то же время избежать слишком простого набора цифр в множителе.

Предположим, $m = 2^{35}$, и операция умножения ускоряется по мере уменьшения количества «единичных» битов в множителе. Можно порекомендовать (экспериментально) такой множитель, как $2^{23} + 2^{14} + 2^2 + 1$. Составляющая 2^{23} делает множитель достаточно большим. Составляющая 2^2 предусматривает высокую степень. Единица необходима для получения максимального периода, и 2^{14} добавляется, чтобы не сделать множитель слишком простым для получения достаточно случайной последовательности. Член, аналогичный 2^{34} , здесь оказался не так хорош, как 2^{23} , поскольку в произведении $2^{34} \cdot X_n$ используется только младший разряд числа X_n (который не так уж и случаен). Если скорость умножения не ограниче-

на, то «случайный» множитель (например, 3 141 592 621), вероятно, будет гораздо более приемлемым.

В действительности определение степени дает лишь один из критериев выбора множителя, к которому добавляют еще несколько.

Рассмотренный метод характеризуется исключительной скоростью, возможностью выбора длины периода, нечувствительностью к выбору исходного числа, прохождением статистических тестов на однородность и, наконец, очень хорошим согласием явлений, моделируемых с использованием полученных случайных чисел этим методом с практикой.

Другие методы

Линейные конгруэнтные последовательности – не единственные возможные генераторы случайных чисел на компьютерах. Одно из распространенных заблуждений заключается в том, что достаточно взять хороший генератор случайных чисел и произвольно модифицировать его, чтобы получить «еще более случайную» последовательность. Как правило, это не так! Например, мы знаем, что из формулы

$$X_{n+1} = (a \cdot X_n + c) \bmod m$$

мы можем получить достаточно хорошие случайные числа. Может ли последовательность

$$X_{n+1} = ((a \cdot X_n) \bmod (m+1) + c) \bmod m$$

быть еще более случайной? Такие предположения рушатся при отсутствии какой-либо теории выбора последовательности, т.е. мы приходим к обобщению генераторов в форме $X_{n+1} = f(X_n)$ со случайно выбранной функцией f . Анализ показывает, что такие последовательности ведут себя не так хорошо, как линейно-конгруэнтный метод, если функция не была тщательно выбрана.

Можно обобщить линейный конгруэнтный метод, преобразовав его в квадратично-конгруэнтный метод:

$$X_{n+1} = (sX_n^2 + aX_n + c) \bmod m.$$

Для случая, когда m – степень двойки, Дж.Х. Конвей предложил квадратичный метод

$$X_0 \bmod 4 = 2, X_{n+1} = X_n(X_n + 1) \bmod 2^e, n \geq 0.$$

Эта последовательность связана с методом средних квадратов Дж. фон Неймана, но дает более длительный период.

Период линейно-конгруэнтного метода велик. Обычно по мере приближения m к размеру машинного слова мы получаем периоды порядка 109 и больше. В большинстве задач используется лишь часть последовательности. При этом размер периода влияет на степень случайности при его достижении в последовательности. Вот почему у нас обычно требуются большие периоды, и разработано множество методов. В одном из них вместо простой зависимости только от X_n вводится зависимость X_{n+1} от X_n и X_{n-1} . Тогда длину периода можно увеличить до m^2 , поскольку последовательность не начнет повторяться до тех пор, пока не выполнится равенство $(X_{n+\lambda}, X_{n+\lambda+1}) = (X_n, X_{n+1})$.

Метод Фибоначчи

Простейший случай зависимости X_{n+1} от более чем одного предыдущего значения реализуется с использованием ряда Фибоначчи:

$$X_{n+1} = (X_n + X_{n-1}) \bmod m.$$

С использованием двоичных компьютеров можно получить периоды $1,5 \cdot m$. При использовании десятичных дробей, если m – простое число, период будет примерно равен $2 \cdot m$. Однако тесты показывают, что числа, полученные из выражения Фибоначчи, не являются достаточно случайными. Поэтому сегодня этот генератор интересен только тем, что он не зависит от типа компьютера.

Один из способов повысить случайность генератора Фибоначчи – использовать

$$X_{n+1} = (X_n + X_{n-k}) \bmod m,$$

где k – достаточно большое число. При удачном выборе X_0, X_1, \dots, X_k эта формула может стать генератором хороших случайных чисел. На первый взгляд, соотношение кажется не очень подходящим для использования на компьютере, поскольку требует запоминания последовательности из k чисел. Однако существует эффективная процедура ее реализации.

Такой генератор обычно работает быстрее, чем предыдущие методы, поскольку он не использует умножения.

Эмпирически установили, что m необходимо выбирать так, чтобы $2 \cdot m$ отображалось в компьютере без обрезки. Для «-битной» мантиссы это должно быть

$$m \leq 2^{l-1}.$$

К сегодняшнему дню такой генератор изучен слабо. Прежде, чем использовать его, необходимо определить показатели случайности генерируемых чисел и провести исследование для разных значений k и X_0, X_1, \dots, X_k . Длина периода больше m .

Существует аналогичный, но гораздо более эффективный способ улучшить случайность линейных конгруэнтных последовательностей, если m – простое число. Например, m можно выбрать как наибольшее простое число, которое может вписаться в машинное слово.

Другой важный класс методов сводится к комбинации генераторов случайных чисел для получения «еще более случайных» последовательностей. Всегда найдутся скептики, которые считают, что линейный конгруэнтный метод, аддитивные методы и т.д. слишком простые, чтобы получить достаточное количество случайных значений. В настоящее время существуют эффективные методы получения из двух наборов значений еще одного с еще лучшими характеристиками случайности.

Метод Макларена – Марсальи

Предположим, у нас есть две последовательности X_0, X_1, \dots и Y_0, Y_1, \dots – случайные числа, распределенные между нулем и $m-1$,

полученные двумя разными способами. Одно из предложений сводится к тому, что числа следует расположить парами по модулю m , получив последовательность

$$Z_n = (X_n + Y_n) \bmod m.$$

В этом случае желательно, чтобы длины периодов этих двух рядов были взаимно простыми числами.

Метод, предложенный Д. Маклареном и Дж. Марсальи, еще лучше и отлично приспособлен для реализации на ЭВМ.

Метод позволяет получить большой период, если периоды отдельных последовательностей взаимно просты³. Даже если длина периода не особенно важна, соседние члены последовательности практически некоррелированы. Это связано с тем, что этот метод лучше ранее рассмотренных, он обеспечивает достаточную случайность из невырожденных последовательностей X и Y .

Авторы этого метода использовали, в частности, следующие строки:

$$X_{i+1} = (2^{17} + 3) \cdot X_i \bmod 2^{35},$$

$$Y_{i+1} = (2^7 + 1) Y_i + 1 \bmod 2^{35}$$

с начальными значениями $X_0 = 1$ и $Y_0 = 0$.

Последовательность, полученная алгоритмом, будет удовлетворять практически любому критерию случайности для последовательностей, генерируемых компьютером.

Б. Гебарт показал, что алгоритм позволяет получать удовлетворительные результаты, даже если его применять к таким неслучайным последовательностям, как Фибоначчи с $X_n = F_{2n} \bmod m$ и $Y_n = F_{2n+1} \bmod m$.

Кроме того, применяется алгоритм Макларена и Марсали, согласно которому сгенерированные случайные числа сначала поме-

³ Взаимно простые числа – целые числа, не имеющие никаких общих делителей, кроме ± 1 .

щаются в специальную матрицу, из которой случайным образом извлекаются случайные элементы. Для этого используются два генератора, один из которых заполняет матрицу случайными числами, а другой случайным образом берёт число из матрицы для дальнейшего использования. Первый генератор сразу же помещает на место новое случайное число. Время генерации случайного числа с помощью этой процедуры примерно в 2 раза больше, чем время с использованием конгруэнтного метода.

В научно-технической литературе последовательности, полученные детерминированным способом, называют псевдослучайными или квазислучайными. Называя их просто случайными последовательностями, подразумеваем, что они лишь создают впечатление случайности. Конечно, все, что можно сказать о случайной последовательности, – это то, что она «случайна по виду». Случайные числа, полученные детерминистическими методами, оказываются пригодными практически для всех приложений.

3.3. Другие виды случайных распределений

Выше были описаны алгоритмы для генерации последовательности чисел U_0, U_1, U_2, \dots , которые ведут себя так, как будто они равномерно распределены между нулем и единицей. Стохастические характеристики ряда реальных систем обычно не могут быть описаны равномерным распределением. Гораздо более распространены другие теоретические распределения, такие как экспоненциальное, нормальное, пуассоновское. Чтобы ввести в модель стохастические характеристики, нужен генератор, который выдает числа с необходимым распределением. Например, если необходимо сделать выбор из k вариантов, то можно использовать случайные целые числа от 1 до k . Если же необходимы случайные интервалы времени между некоторыми независимыми событиями, то необходимо использовать случайные величины с экспоненциальным распределением.

Разные распределения случайных величин можно получить, используя случайно распределенные числа U_0, U_1, U_2, \dots . Для этого

существует ряд косвенных методов, изучение которых помогает понять, как случайные числа можно использовать в приложениях моделирования.

Примем, что U_0, U_1, U_2, \dots – это случайная последовательность действительных чисел, равномерно распределенных между нулем и единицей. Буквой U без индекса будем обозначать текущий элемент этой последовательности. При этом необходимо учитывать, что число в компьютере отображается с ограниченной точностью.

Многие из наиболее известных методов получения случайных величин для заданных распределений были первоначально предложены Дж. Фон Нейманом в 1950-х годах, а затем постепенно усовершенствованы другими авторами, особый вклад в их усовершенствование внес Джордж Марсалис.

Моделирование непрерывных случайных величин

Дальнейшие рассуждения будем производить, опираясь на следующую теорему.

Теорема 3.5. *Если мы рассматриваем случайное число U как значение оси ординат (значение случайной величины U , отложенное по вертикальной оси координат) и находим такое значение случайной величины H , для которого $F(X) = U$, то получаем значения случайной величины H , функция распределения которой равна $F(x)$.*

Нормальное распределение

Стандартное нормальное распределение (со средним значением, равным нулю, и стандартным отклонением, равным единице, $N(0,1)$), представляет собой одно из наиболее важных неравномерных непрерывных распределений:

$$F(X) = \frac{1}{\sqrt{2 \cdot \pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt.$$

Если H имеет такое распределение, то функция распределения случайной величины

$$Y = \mu + \sigma \cdot x$$

определяет нормальное распределение со средним значением μ и стандартным отклонением σ .

Метод на основе центральной предельной теоремы

Данный метод является наиболее широко используемым.

Теорема 3.6. Для случайной выборки (X_1, X_2, \dots, X_n) со средним значением μ и стандартным отклонением σ распределение выборочного среднего значения X_n представляет собой случайную величину, распределение которой стремится к нормальному распределению со средним значением μ и дисперсией $\frac{\sigma^2}{n}$, когда $n \rightarrow \infty$,

т.е. функция распределения случайной величины $\frac{(\bar{X}_n - \mu)}{(\sigma / \sqrt{n})}$ или

$\frac{(\sum_{i=1}^n X_i - n \cdot \mu)}{(\sigma / \sqrt{n})}$ стремится к нормальной функции распределения.

Рассмотрим последовательность случайных чисел (X_i) , которая имеет равномерное распределение с $\mu = 1/2$ и $\sigma^2 = \frac{1}{12}$. Из последовательности $(X_i; i = 1, 2, \dots, n)$ можно определить случайную величину

$$Y = \frac{S_n - n\mu}{\sigma\sqrt{n}} = \frac{\sum_{i=1}^n X_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}},$$

которая, когда n достаточно велико, стремится к нормальному распределению.

Выбор размера выборки, т.е. числа n , определяется в соответствии с требованием затрачивать как можно меньше компьютерного времени, поскольку очень важно, чтобы генераторы случайных чисел были чрезвычайно быстрыми! Предполагая, что $n = 12$ достаточно большое число, чтобы получить хорошее приближение к нормально распределенной переменной, получим

$$Y = \sum_{i=1}^{12} X_i - 6.$$

В результате получилась простая и быстрая процедура, но недостатком является то, что значения случайной величины Y , сгенерированные указанным выше способом, находятся в ограниченной области вокруг среднего значения ($\mu = \pm 6$), а «хвосты» распределения за их пределами никогда не появляются.

Метод полярных координат

Рассмотрим алгоритм, который генерирует две независимые нормально распределенные случайные величины X_1 и X_2 на основе двух заданных независимых случайных чисел U_1 и U_1 .

Шаг 1 (генерация случайных чисел). Сгенерируем два независимых случайных числа U_1, U_2 методом получения равномерно распределенных чисел между нулем и единицей. Присвоим $V_1 = 2U_1 - 1$. Теперь V_1 и V_2 принимают равномерно распределённые значения между -1 и $+1$. На большинстве ЭВМ на этом этапе удобнее всего представлять V_1 и V_2 в форме с плавающей запятой.

Шаг 2 (расчет значения S). Примем $S = V_1^2 + V_2^2$.

Шаг 3 (проверяем будет ли значение $S \geq 1$?). Если $S \geq 1$, переходим на шаг 1.

Шаг 4 (расчет значений X_1 и X_2). Значения переменных X_1, X_2 определяем по формулам

$$X_1 = V_1 \cdot \sqrt{\frac{-2 \cdot \ln S}{S}}, \quad X_2 = V_2 \cdot \sqrt{\frac{-2 \cdot \ln S}{S}}.$$

Эти значения и есть искомые значения, соответствующие нормальному распределению случайной величины.

Для того чтобы доказать корректность метода, воспользуемся методами аналитической геометрии.

Если на шаге 3 $S < 1$, точка поверхности с прямоугольными координатами (V_1, V_2) представляет собой случайную точку, равномерно распределенную внутри единичного круга. Переходя к полярным координатам $V_1 = R \cdot \cos \theta$, $V_2 = R \cdot \sin \theta$, находим $S = R^2$, $X_1 = \sqrt{-2 \cdot \ln S} \cdot \cos \theta$, $X_2 = \sqrt{-2 \cdot \ln S} \cdot \sin \theta$. Используя полярные ко-

ординаты $X_1 = R' \cdot \cos \theta'$, $X_2 = R' \cdot \sin \theta'$, если $\theta' = \theta$ и $R' = \sqrt{-2 \cdot \ln S}$, то они независимы, поскольку R и θ независимы внутри единичного круга. Кроме того, значения θ' равномерно распределены между 0 и 2π , и вероятность того, что $R' \leq r$ равна вероятности события $-2 \cdot \ln S \leq r^2$, т.е. вероятности событий $S \geq e^{-\frac{r^2}{2}}$. Последнее можно привести к виду $1 - e^{-\frac{r^2}{2}}$, если $S = R^2$ и значения будут равномерно распределены между нулем и единицей. Вероятность того, что R' лежит между r и $r + dr$, равна дифференциалу $1 - e^{-\frac{r^2}{2}}$ и равна $re^{-\frac{r^2}{2}} dr$. Аналогично найдем вероятность θ' в интервале между θ и $\theta + d\theta$, которая будет равна $(\frac{1}{2\pi})d\theta$. Следовательно, вероятность того, что $X_1 \leq x_1$ и $X_2 \leq x_2$ равна

$$\int_{\{(r, \theta) | r \cdot \cos \theta \leq x_1, r \cdot \sin \theta \leq x_2\}} \frac{1}{2\pi} e^{-\frac{r^2}{2}} r dr d\theta = \frac{1}{2\pi} \int_{\{(x, y) | x \leq x_1, y \leq x_2\}} e^{-\frac{(x^2 + y^2)}{2}} dx dy.$$

Это доказывает, что X_1 и X_2 независимы и нормально распределены, как и должно быть. Описанный метод был предложен Дж. Боксом и М. Малером [74].

Метод Тейчроева [75]

Сгенерируем 12 независимых случайных чисел U_1, U_2, \dots, U_{12} , равномерно распределенных между нулем и единицей. Введем

$$R = \frac{U_1 + U_2 + \dots + U_{12} - 6}{4}.$$

Посчитаем

$$X = \left(\left(\left(\left(a_9 \cdot R^2 + a_7 \right) \cdot R^2 + a_5 \right) \cdot R^2 + a_3 \right) \cdot R^2 + a_1 \right) \cdot R,$$

где $a_1 = 3,949846138$, $a_3 = 0,252408784$, $a_5 = 0,076542912$, $a_7 = 0,008355968$, $a_9 = 0,029899776$.

Получаемые таким образом X будут хорошим приближением к нормально распределённым случайным числам. При использовании приведенного выражения никогда не получаются очень большие значения X . Лишь с вероятностью менее $\frac{1}{50000}$ генерируются числа, выходящие за пределы стандартного отклонения.

Метод основан на том, что R имеет приблизительно нормальное распределение со средним значением 0 и стандартным отклонением $\frac{1}{4}$. Пусть $F_1(X)$ – вещественное распределение R , а $F(x)$ – нормальное распределение. Положим $X = F^{-1}(F_1(R))$; поскольку $F_1(R)$ – равномерно распределенная случайная величина, X будет иметь нормальное распределение. Формула, приведенная выше для X , представляет собой аппроксимацию функции $F^{-1}(F_1(R))$ полиномом на интервале $|R| \leq 1$.

Из рассмотренных трех методов генерации нормальных случайных величин метод полярных координат довольно медленный, но обеспечивает абсолютную точность. Метод Д. Тейчроева при реализации на ЭВМ требует меньше всего машинной памяти. Этот метод является приближительным, хотя для большинства приложений он дает точность (погрешность которой не превышает $2 \cdot 10^{-4}$ при $|R| \leq 1$).

Многokратное использование нормально распределенных случайных величин требует большого количества случайных чисел, поэтому скорость их генерации очень важна.

Экспоненциальное распределение

Другой важной формой случайных величин являются экспоненциально распределенные переменные. Такие случайные величины нужны в первую очередь в задачах, где соблюдается «время появления». Например, если радиоактивное вещество испускает

в среднем каждые $\frac{1}{\lambda}$ секунды одну альфа-частицу, то временные интервалы между двумя последовательными выбросами частиц имеют экспоненциальное распределение со средним значением $\frac{1}{\lambda}$. Такое распределение определяется функцией

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

Отсюда следует, что если X имеет экспоненциальное распределение со средним значением 1, то $\frac{X}{\lambda}$ имеет экспоненциальное распределение со средним значением $\frac{1}{\lambda}$. Следовательно, достаточно рассмотреть случай $\lambda = 1$.

Логарифмический метод

Функцию $y = F(x) = 1 - e^{-\lambda x}$ можно представить в виде $x = F^{-1}(y) = -\frac{1}{\lambda} \ln(1 - y)$. Величина $-\ln(1 - U)$ имеет стандартное показательное распределение. Поскольку $1 - U$ является стандартным равномерно распределенным, если U является стандартным равномерно распределенным случайным числом, то случайная величина

$$X = -\left(\frac{1}{\lambda}\right) \ln U$$

имеет показательное распределение со средним значением, равным $\frac{1}{\lambda}$. Следует обратить внимание, что при программной реализации следует избегать случая $U = 0$.

Метод случайной минимизации

Приведенный ниже алгоритм, разработанный Дж. Марсалисом, вычисляет значения для экспоненциально распределенной случай-

ной величины без использования логарифма со средним значением, равным 1. Алгоритм опирается на использование таблицы констант $P(j)$, $Q(j)$ для $j \geq 1$, которые определяются по формулам

$$P(j) = 1 - \frac{1}{e^j}, \quad Q(j) = \frac{1}{e-1} \left(\frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{j!} \right).$$

Длина составляемых таблиц ограничена значением максимального дробного члена, который может поместиться в машинное слово.

Шаг 1 (начало первой части). Присваиваем значение $j = 1, X = 0$. Генерируем случайное число U и присваиваем дробной части X значение U .

Шаг 2 (проверка на минимизацию). Если $U < Q(j)$, то переходим на шаг 4.

Шаг 3 (минимизация). Присваиваем $j = j + 1$. Генерируем случайное число $U \rightarrow$ и присваиваем дробной части X значение U . Переходим на шаг 2.

Шаг 4 (начало второй части). Проведенные на предыдущих шагах операции позволили вычислить дробную часть искомого числа X . Теперь необходимо определить целую часть. Для этого сгенерируем новое случайное число U и присвоим $j = 1$.

Шаг 5 (проверяем на необходимость поправки). Если $U < P(j)$, алгоритм завершается.

Шаг 6 (корректировка). Присваиваем $j = j + 1, X = X + 1 \rightarrow$ и возвращаемся к шагу 5.

Чтобы показать корректность метода, проанализируем распределение X к началу шага 4. Если n – последнее значение, которое принимает j , то $X = \min(U_1, U_2, \dots, U_n)$, где U_1, U_2, \dots, U_n – независимые случайные числа. Отсюда вероятность того, что $X \leq x$ равна $p_n(x) = 1 - (1 - x)^n$. Вероятность того, что n является окончательным

значением, равна $Q(n) - Q(n-1) = \frac{1}{(e-1)n!}$. Следовательно, полное

значение события $X \leq x$ равно

$$\sum_{n \geq 1} \frac{P_n}{(e-1) \cdot n!} = \frac{e}{(e-1)} (1 - e^{-x}), 0 \leq x \leq 1.$$

Аналогично, рассматривая шаги 4–6, находим, что $|X| \leq m$ с вероятностью $P(m+1)$, а вероятность того, что $m \leq X \leq m+x$ будет

$$(P(m+1) - P(m)) \left(\frac{e}{e-1} (1 - e^{-x}) \right) = e^{-m} - e^{-(m+x)}, 0 \leq x \leq 1.$$

Это доказывает, что X имеет распределение

$$F(x) = 1 - e^{-x} \text{ для } 0 \leq x < \infty.$$

Распределение Эрланга

В процессах обслуживания время обслуживания обычно имеет экспоненциальное распределение, но часто его лучше описать распределением Эрланга, определяемым функцией плотности вероятности

$$f(x) = \frac{\lambda(\lambda x)^{k-1}}{(k-1)!} e^{-\lambda x}; \quad x > 0; \quad \lambda > 0; \quad k \in N.$$

При $k = 1$ распределение Эрланга сводится к экспоненциальному.

Можно показать, что распределение Эрланга порядка k получается как плотность распределения суммы k независимых экспоненциально распределенных случайных величин

$$X = \sum_{i=1}^k X_i = -\frac{1}{\lambda} \ln \prod_{i=1}^k U_i.$$

Распределение χ^2

Распределение χ^2 с n степенями свободы, которое также называют гамма-распределением порядка $\frac{n}{2}$, определяется выражением

$$F(x) = \frac{1}{2^{\frac{n}{2}} \cdot \Gamma\left(\frac{n}{2}\right)} \int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt, \quad x \geq 0.$$

Если $n = 2k$, где k – целое число, то получим $X = 2(Y_1 + Y_2 + \dots + Y_k)$, где Y_i – независимые случайные величины с экспоненциальным распределением со средним значением, равным $1 \forall i$.

Если $n = 2k + 1$, то получим $X = 2(Y_1 + Y_2 + \dots + Y_k) + Z^2$, где Z – независимая нормально распределенная случайная величина со средним значением, равным 0 , и значением дисперсии 1 .

При этом, если Y_1, \dots, Y_k находятся логарифмическими методами, то для определения $Y_1 + \dots + Y_k = -\ln(U_1 \cdot \dots \cdot U_k)$ необходимо вычислить только один логарифм.

Моделирование дискретных случайных величин

Самый простой и общий вид распределения, необходимый для практических приложений – это распределение вероятностей случайных величин, принимающих только целые значения.

Чтобы получить случайное целое число X от 0 до $n - 1$, мы можем умножить U на n и принять его равным

$$X = \lfloor n \cdot U \rfloor.$$

Если, с другой стороны, требуется случайное целое число от 1 до n , мы добавляем к предыдущему результату 1 . Этот метод дает каждое целое число в указанном интервале с равной вероятностью.

В общем случае мы можем присваивать разные целые числа неравного веса. Предположим, что значение $X = x_1$ необходимо получать с вероятностью p_1 , $X = x_2$ с вероятностью p_2 , ... и $X = x_n$ с вероятностью p_n . Итак, мы хотим получить (смоделировать) значения случайной величины X , распределение вероятностей которой определяется выражением

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix} \sum_{i=1}^n p_i = 1.$$

Можно сгенерировать стандартное равномерно распределенное число U и использовать выражение

$$X = \begin{cases} x_1, & \text{если } 0 \leq U < p_1, \\ x_2, & \text{если } p_1 \leq U < p_1 + p_2, \\ \dots & \\ x_n, & \text{если } p_1 + p_2 + \dots + p_{n-1} \leq U < 1. \end{cases}$$

Правильность этой процедуры легко доказать. Давайте выделим интервал $0 < y < 1$ и разделим его на n частей, длины которых по порядку равны p_1, p_2, \dots, p_n . Координаты отдельных точек будут равны

$$y_1 = p_1, y_2 = p_1 + p_2, \dots, y_n = 1 - p_n.$$

Полученные интервалы обозначим $\Delta_1, \Delta_2, \dots, \Delta_n$.

Теперь мы сгенерируем значение U и изучим точку $y = U$. Если эта точка попадает в интервал Δ_i , то мы считаем, что значение $X = x_i$ было реализовано в результате этого моделирования. Основанием для этого утверждения является то, что поскольку случайная величина U равномерно распределена на интервале $(0, 1)$, вероятность того, что U будет найдена на некотором интервале, равна длине этого интервала, что означает:

$$P(0 \leq U < p_1) = P(U \in \Delta_1) = p_1,$$

$$P(p_1 \leq U < p_1 + p_2) = P(U \in \Delta_2) = p_2,$$

...

$$P(p_1 + p_2 + \dots + p_{n-1} \leq U < 1) = P(U \in \Delta_n) = p_n.$$

Поскольку событие $X = x_i$ эквивалентно событию

$$p_1 + p_2 + \dots + p_{i-1} \leq U < p_1 + p_2 + \dots + p_i,$$

то отсюда следует, что их вероятности равны

$$P(X = x_i) = P(U \in \Delta_i) = p_i.$$

При этом мы также доказали следующую теорему:

Теорема 3.7. *Случайная величина X определяется по формуле $X = x_i$, когда $U \in \Delta_i$ имеет распределение вероятностей:*

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}.$$

Некоторые из дискретных распределений имеют большое практическое значение и заслуживают отдельного рассмотрения.

Геометрическое распределение

Если событие имеет вероятность появления p , то число (N) независимых испытаний, необходимых для того, чтобы событие произошло (или количество испытаний среди событий), подчиняется геометрическому распределению.

Если $N = 1$ с вероятностью p , $N = 2$ с вероятностью $(1-p) \cdot p$, ..., $N = n$ с вероятностью $(1-p)^{n-1} p$, то это не соответствует геометрическому распределению.

Для генерации значения случайной величины, когда p мало, следует воспользоваться следующей формулой

$$N = \lceil \frac{\ln U}{\ln(1-p)} \rceil.$$

Чтобы это проверить, убедимся, что $\lceil \frac{\ln U}{\ln(1-p)} \rceil = n$ тогда

и только тогда, когда $n-1 < \frac{\ln U}{\ln(1-p)} \leq n$, т.е. $(1-p)^{n-1} > U > (1-p)^n$

с вероятностью $p(1-p)^n$, что и следовало получить.

Особый случай $p = \frac{1}{2}$ еще проще смоделировать на двоичной машине, поскольку исходная формула приводится к виду $N = -\ln U$, т.е. N на единицу больше, чем количество первых нулевых классов в двоичном представлении U .

Биномиальное распределение

Если событие происходит с вероятностью p , и мы проводим n независимых экспериментов, то вероятность того, что событие произойдет k раз, равна

$$p_k = P(x = k) = f(k) = \binom{n}{k} p^k (1-p)^{n-k}, 0 < p < 1, k = 0, 1, 2, \dots, n.$$

Моделирование значения случайной величины X , распределение которой является вероятностным

$$X = \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots & n \\ p_0 & p_1 & p_2 & \dots & p_k & \dots & p_n \end{pmatrix}.$$

Добиться этого можно следующим образом:

если $0 \leq U < p_0$, то $X = 0$,

если $p_0 \leq U < p_0 + p_1$, то $X = 1$,

...

если $\sum_{i=0}^k p_i \leq U < \sum_{i=0}^{k+1} p_i$, то $X = k + 1$.

При реализации на ЭВМ такой подход работает достаточно медленно. При больших n более эффективно использовать аппроксимацию для формирования биномиального распределения случайной величины. Если n велико, а p мало и выполняется $\lambda = np = \text{const}$, то биномиальное распределение стремится к Пуассоновскому, поэтому следует применить методы построения Пуассоновского распределения. Кроме того, когда n велико, биномиальное распределение приблизительно описывается нормальным распределением с математическим ожиданием np и стандартным отклонением $\sqrt{np(1-p)}$. В этом случае биномиальная переменная получается с помощью стандартного нормального распределения как

$$X(p, n) = \text{int}\left(\sqrt{np(1-p)} \cdot N(0, 1) + np\right).$$

Пуассоново распределение

Распределение Пуассона характеризует количество событий в единицу времени, каждое из которых может произойти в произвольный момент. Под полностью случайным событием мы понимаем события, появление которых не зависит от предыдущих. Это означает, что время между двумя последовательными событиями, каким бы коротким или длинным оно ни было, не влияет на время наступления следующего события. Пусть средняя интенсивность вступлений в единицу времени равна λ , тогда $\frac{1}{\lambda}$ – среднее время между двумя соседними событиями. Вероятность того, что k событий произойдет в единицу времени, описывается распределением Пуассона

$$p_k = P(X = k) = f(k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, 2, \dots, \lambda > 0.$$

Моделирование значения случайной величины X , распределение которой является вероятностным

$$X = \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots & n \\ p_0 & p_1 & p_2 & \dots & p_k & \dots & p_n \end{pmatrix}.$$

Добиться этого можно следующим образом:

если $0 \leq U < p_0$, то $X = 0$,

если $p_0 \leq U < p_0 + p_1$, то $X = 1$,

...

если $\sum_{i=0}^k p_i \leq U < \sum_{i=0}^{k+1} p_i$, то $X = k + 1$.

Генерация случайных величин с распределением Пуассона и параметром λ , основанная на приведенном выше алгоритме, работает достаточно медленно. В условиях, когда требуется высокая скорость, он практически непригоден! Для практических приложе-

ний гораздо лучше себя показывают другие методы генерации распределения Пуассона.

Экспоненциальный метод

Для получения распределения Пуассона с произвольным λ необходимо выполнить следующие шаги:

Шаг 1 (определение показателя экспоненты). Выполняем присваивание $p = e^{-\lambda}$, $X = 0$, $lj = 1$. Несмотря на то что $e^{-\lambda}$ обычно вычисляется с использованием арифметики с плавающей запятой, можно сократить время выполнения операции, правильно выбрав значения и округлив p и lj .

Шаг 2 (получение случайного числа). Сгенерировать случайное число U , равномерно распределенное между 0 и 1.

Шаг 3 (умножение). Присвоим $lj = lj \cdot U$.

Шаг 4 (проверка значения $e^{-\lambda}$). Если $lj \geq p$, присваиваем $X = X + 1$ и возвращаемся к шагу 2. В противном случае выводим X и завершаем работу алгоритма.

Для доказательства корректности метода заметим, что независимые равномерно распределенные случайные величины удовлетворяют условиям

$$U_1 \geq p, U_1 U_2 \geq p, \dots, U_1 U_2 \dots U_n \geq p, U_1 U_2 \dots U_{n+1} < p$$

для значения вероятности

$$\frac{p \left(\ln \left(\frac{1}{p} \right) \right)^n}{n!} \text{ для } 0 < p \leq 1.$$

Применяя метод индукции к n , путем интегрирования

$$\int_p^1 \frac{p(u_1) \left(\ln \left(\frac{u_1}{p} \right) \right)^{n-1}}{(n-1)!} du_1$$

мы получаем требуемый результат.

Метод суммирования (метод П. Крибса)

Другой метод получения случайных величин с распределением Пуассона основан на свойстве суммирования. Алгоритм поиска чисел с распределением Пуассона с произвольным λ использует вспомогательную таблицу $M(1) < M(2) < \dots < M(n)$ и описан ниже.

Шаг 1 (первоначальная настройка). Присвоим $m = \lambda$, $j = n$, $N = 0$.

Шаг 2: (проверка условия $m \geq M(j)$) Если $m < M(j)$, перейдем к шагу 5.

Шаг 3 (генерация среднего арифметического $M(j)$). Возьмем значение X случайной целочисленной переменной, которая имеет распределение Пуассона со средним значением $M(j)$.

Шаг 4 (изменение N, m). Присвоим $m = m - M(j)$, $N = N + X$ и вернемся к шагу 2.

Шаг 5 (уменьшение j). Уменьшаем j на 1. Если $j > 0$, то переходим на шаг 2, в противном случае алгоритм завершается.

Чтобы использовать этот алгоритм, необходимо составить специальные таблицы $M(1), M(2), \dots, M(n)$ для частичных значений λ . Например, мы можем взять $n = 10$, тогда

$j =$	1	2	3	4	5	6	7	8	9	10
$(M)j =$	2^{-15}	2^{-12}	2^{-9}	2^{-6}	2^{-3}	2^{-1}	1	2	4	8

Этот метод неэффективен при больших значениях λ , скажем, когда $\lambda \geq 50$. При $\lambda < M(1) = 2^{-12}$ описанный алгоритм присваивает $N = 0$, поскольку вероятность того, что $N > 0$ равна $1 - e^{-\lambda}$, т.е.

меньше $\frac{1}{32000}$. Распределение Пуассона для малых значений λ гораздо проще, поскольку для практических целей N будет достаточно малым. Только распределение с $M(j) = 4$ и 8 из приведенной выше таблицы значений нуждается в таблицах большего размера при выполнении шага 3.

Такой метод предложил П. Крибс.

Генерация случайных чисел основана на генерации равномерно распределенных независимых случайных числах. В компьютерном моделировании последовательность случайных чисел не является случайной последовательностью, поскольку она является результатом детерминированного алгоритма и в лучшем случае лишь аппроксимирует равномерно независимо распределенные случайные величины. К счастью, во многих приложениях метода Монте-Карло необходимы лишь отдельные свойства однородности и независимости последовательности случайных чисел.

Поскольку генератор псевдослучайных чисел не может идеально приближаться к реальному генератору случайных чисел, его следует выбирать так, чтобы он максимально соответствовал свойствам, важным для каждой конкретной задачи. Важно, чтобы при проверке ошибок округления разряды случайных чисел были распределены случайным образом, особенно те, которые находятся в конце числа.

Качественный генератор случайных чисел обладает следующими свойствами:

- а) скорость генерации таких чисел того же порядка, что и скорость работы компьютера;
- б) программа занимает всего несколько ячеек памяти;
- в) существует возможность повторной генерации того же случайного числа / последовательности;
- г) проверять «качество» такой последовательности случайных чисел необходимо только один раз применительно к решению смежных задач.

3.4. Проверка принадлежности генератора случайных чисел и результатов моделирования чисел к заданному распределению

Теперь рассмотрим тесты для сгенерированных псевдослучайных последовательностей. Однако следует иметь в виду, что на сегодняшний день отсутствует адекватная научная база для указан-

ных ниже процедур тестирования. Причиной этого является тот факт, что мы имеем дело с тестами, предназначенными для случайных процессов, которые мы теперь хотим применить к псевдослучайным процессам, чтобы доказать их свойства случайности. Принято считать, что тестирование генераторов и поиск тех, которые удовлетворяют требованиям случайности и однородности сгенерированной последовательности, является необходимым и основным предположением о правильности результатов.

Тестирование генераторов псевдослучайных чисел связано с единообразием и независимостью, и тестирование проводится отдельно. В то же время единообразие гораздо проще обеспечить и доказать. В рамках теории статистики существуют надежные тесты на однородность (такие как критерий хи-квадрат или критерий Колмогорова – Смирнова). Независимость гораздо труднее проверить из-за большого количества различных свойств, которые могут потребоваться от статистически независимой последовательности.

В учебниках по статистике можно найти большое число примеров, когда на основании наборов данных строятся их эмпирические законы распределения и/или гистограммы распределения, на основании которых делается вывод о приближении к тому или иному закону распределения.

Существует два типа тестов: эмпирические тесты, когда компьютер манипулирует группами порядковых номеров и дает оценку по определенным статистическим критериям, и теоретические тесты, когда мы находим некоторые характеристики последовательности, используя методы теории чисел, на основе рекуррентных отношений, по которым была сгенерирована последовательность.

Универсальные тесты для анализа случайных рядов

Тест χ^2 (хи-квадрат)

Критерий χ^2 (хи-квадрат) является, вероятно, наиболее широко используемым из всех статистических тестов. Он используется не только как отдельный тест, но и как неотъемлемая часть многих других тестов. Тест можно применять как в непрерывном, так и в дискретном случаях.

В 1900 году К. Пирсон ввел меру χ^2 для измерения отклонения между эмпирическими и ожидаемыми (предполагаемыми) теоретическими частотами:

$$\chi^2 = \sum_{i=1}^r \frac{(f_i - f_i^{(t)})^2}{f_i^{(t)}},$$

где f_i – частота «-го» значения случайной величины в выборке или частота i -го класса; $f_i^{(t)}$ – соответствующая теоретическая частота; r – количество классов/значений, $N = \sum f_i = \sum f_i^{(t)}$ – размер выборки, количество независимых экспериментов.

Диапазон случайных чисел разбивается на r интервалов, которые не обязательно должны быть равны друг другу, но чаще всего равны. Определяются частоты сгенерированных чисел, попадающих в каждый отдельный интервал. После этого рассчитываются теоретические частоты ($f_i^{(t)}$) для распределения, которое мы предполагаем иметь.

Применяем критерий χ^2 для проверки гипотезы о соответствии эмпирического и предполагаемого теоретического распределений при условии, что теоретические частоты $f_i^{(t)} \geq 5$. Если $f_i^{(t)} < 5$, что часто бывает в начальном и конечном классах / значениях, то классы / значения включаем в соседние.

Если эмпирические отклонения от теоретических распределений велико, χ^2 становится очень большим.

В табл. 16 приведены для различных значений k (степеней свободы) значения, соответствующие вероятностям α согласно соотношению:

$$P(\chi^2 - \chi_\alpha^2) = \alpha,$$

для $\alpha = 0,99, 0,95, \dots, 0,05, 0,01$.

Вероятность α называется критическим коэффициентом согласия гипотезы между эмпирическим и теоретическим распределением (α – уровень значимости гипотезы или вероятность того, что по-

грешность превзойдёт некоторое предельное критическое решение, а $(1 - \alpha)$ α – доверительная вероятность принятия гипотезы). Число степеней свободы определяем как

$$k = r - l - 1,$$

где r – количество классов / значений, l – количество неизвестных параметров предполагаемого распределения.

Таблица 16

Некоторые процентные точки χ^2 –
распределения [76]

	$\alpha = 0,99$	$\alpha = 0,95$	$\alpha = 0,75$	$\alpha = 0,5$	$\alpha = 0,25$	$\alpha = 0,05$	$\alpha = 0,01$
$k = 1$	0,00016	0,00393	0,1015	0,4549	1,323	3,841	6,635
$k = 2$	0,0201	0,1026	0,5754	1,386	2,773	5,991	9,210
$k = 3$	0,1148	0,3518	1,213	2,366	4,108	7,815	11,34
$k = 4$	0,2971	0,7170	1,923	3,357	5,385	9,488	13,28
$k = 5$	0,5543	1,1455	2,675	4,351	6,626	11,07	15,09
$k = 6$	0,8721	1,635	3,455	5,348	7,841	12,59	16,81
$k = 7$	1,239	2,167	4,255	6,346	9,037	14,07	18,48
$k = 8$	1,646	2,733	5,071	7,344	10,22	15,51	20,09
$k = 9$	2,088	3,325	5,899	8,343	11,39	16,92	21,67
$k = 10$	2,558	3,940	6,737	9,342	12,55	18,31	23,21
$k = 11$	3,053	4,575	7,584	10,34	13,70	19,68	24,72
$k = 12$	3,571	5,226	8,438	11,34	14,85	21,03	26,22
$k = 15$	5,229	7,261	11,04	14,34	18,25	25,00	30,58
$k = 20$	8,260	10,85	15,45	19,34	23,83	31,41	37,57
$k = 30$	14,95	18,49	24,48	29,34	34,80	43,77	50,89
$k = 50$	29,71	34,76	42,94	49,33	56,33	67,50	76,15
$k > 50$	$k + \sqrt{2k} \cdot x_\alpha + \frac{2}{3} x_\alpha^2 - \frac{2}{3} + O\left(\frac{1}{\sqrt{k}}\right)$						
x_α	-2,33	-1,64	-0,674	0,00	0,674	1,64	2,33

Таким образом, число степеней свободы для равномерного распределения равно $k = r - 1$, для биномиального – $k = r - 2$, для нормального распределения – $k = r - 3$, для пуассоновского – $k = r - 2$ и т.д.

Критерий Пирсона

Если:

1) рассчитанное значение χ^2 больше, чем χ_α^2 , то гипотезу отвергаем, считая, что отклонения эмпирического распределения от предполагаемого значительны;

2) вычисленное значение χ^2 меньше, чем χ_α^2 , не является основанием для отклонения гипотезы, что не означает, что гипотеза полностью верна. Применяя другие тесты, мы можем подтвердить исходное предположение.

Тест Романовского

Отклонение эмпирического распределения от предполагаемого теоретического имеет случайный характер, то есть гипотеза принимается, если:

$$\frac{|(\chi^2 - k)|}{\sqrt{2k}} \leq 3,$$

и отклоняется, если:

$$\frac{|(\chi^2 - k)|}{\sqrt{2k}} > 3,$$

где k – число степеней свободы.

Рассмотрим значения с частотным представлением значений, приведенном в табл. 17.

Таблица 17

Частотное представление случайных значений

i	0	1	2	3	4	5	6	7	8	9
f_i	256	241	247	247	244	244	272	238	256	255

Если бы цифры появлялись одинаково часто, мы ожидали бы, что каждая цифра будет встречаться $2500/10 = 250$ раз. Тест χ^2 показывает, что полученные отклонения носят случайный характер:

$$\chi^2 = \sum_{i=0}^9 \frac{(f_i - 250)^2}{250} = 3,584 < \chi_{0.05}^{2(9)} = 16,919.$$

Если выполнить проверку по критерию Пирсона, то $\chi^2 = 8,400 < 16,919 = \chi_{0.05}^{2(9)}$, т.е. у нас нет оснований отвергать гипотезу о соответствии тестируемого распределения теоретическому.

При проверке по критерию Романовского отклонение эмпирического распределения от теоретического носит случайный характер, т.е. гипотеза принимается, потому что $0,141 < 3$.

λ-тест (тест Колмогорова – Смирнова)

В случаях, когда эмпирическое распределение сравнивается с теоретическим непрерывным распределением вероятностей, хорошо подходит для проверки гипотезы об их совпадении критерий Колмогорова. Колмогоров доказал, что абсолютная величина есть максимальное отличие эмпирической функции распределения от предполагаемой теоретической.

$$D_n = \max_{-\infty < X < \infty} |F_n(x) - F(x)|,$$

где n – количество элементов в выборке (случайная величина, имеющая собственное распределение вероятностей). Точнее, он придумал функцию распределения вероятностей случайной величины $D_n \sqrt{n}$.

$$\lim_{n \rightarrow \infty} P(D_n \sqrt{n} < \lambda) = Q(\lambda) = \sum_{k=-\infty}^{+\infty} (-1)^k e^{-2k^2 \lambda^2}.$$

В табл. 18 приведены значения доверительной вероятности $P(\lambda)$ для различных значений $\lambda (\lambda > 0)$.

Таблица 18

Некоторые процентные точки распределений λ [76]

Параметр	$\alpha = 0,99$	$\alpha = 0,95$	$\alpha = 0,75$	$\alpha = 0,5$	$\alpha = 0,25$	$\alpha = 0,05$	$\alpha = 0,01$
$n = 1$	0,01000	0,05000	0,2500	0,5000	0,7500	0,9500	0,9900
$n = 2$	0,01400	0,06749	0,2929	0,5176	0,7071	1,0980	1,2728

Параметр	$\alpha = 0,99$	$\alpha = 0,95$	$\alpha = 0,75$	$\alpha = 0,5$	$\alpha = 0,25$	$\alpha = 0,05$	$\alpha = 0,01$
$n = 3$	0,01699	0,07919	0,3112	0,5147	0,7539	1,1017	1,3589
$n = 4$	0,01943	0,08789	0,3202	0,5110	0,7642	1,1304	1,3777
$n = 5$	0,02152	0,09471	0,3249	0,5245	0,7674	1,1392	1,4024
$n = 6$	0,02336	0,1002	0,3272	0,5319	0,7703	1,1463	1,4144
$n = 7$	0,02501	0,1048	0,3280	0,5364	0,7755	1,1537	1,4246
$n = 8$	0,02650	0,1086	0,3280	0,5392	0,7797	1,1586	1,4327
$n = 9$	0,02786	0,1119	0,3274	0,5411	0,7825	1,1624	1,4388
$n = 10$	0,02912	0,1147	0,3297	0,5426	0,7845	1,1658	1,4440
$n = 11$	0,03028	0,1172	0,3330	0,5439	0,7863	1,1688	1,4484
$n = 12$	0,03137	0,1193	0,3357	0,5453	0,7880	1,1714	1,4521
$n = 15$	0,03424	0,1244	0,3412	0,5500	0,7926	1,1773	1,4606
$n = 20$	0,03807	0,1298	0,3461	0,5547	0,7975	1,1839	1,4698
$n = 30$	0,04354	0,1351	0,3509	0,5605	0,8036	1,1916	1,4801
$n > 30$	$y_p + \frac{1}{6}n^{-\frac{1}{2}} + O\left(\frac{1}{n}\right)$						
$y_p =$	0,07089	0,1601	0,3793	0,5887	0,8326	1,2239	1,5174

Критерий Колмогорова – Смирнова

Для проверки гипотезы о соответствии эмпирического и предполагаемого теоретического распределения с вероятностью $(1 - \alpha)$ находим из таблицы такое значение λ_α , что $p(\lambda_\alpha) = (1 - \alpha)$. Если $D_n\sqrt{n} < \lambda$, у нас нет оснований отвергнуть гипотезу о соответствии между эмпирическим и предполагаемым теоретическим распределением. Если $D_n\sqrt{n} > \lambda_\alpha$, то гипотезу отвергаем (с уровнем значимости α).

Применим критерий Колмогорова – Смирнова, когда

$$D_n\sqrt{n} = 0,800 < \lambda_{0,05}^{(9)} = 1,16,$$

лямбда (5 %) = 1,16 т.е. у нас нет оснований отвергать гипотезу о соответствии между эмпирическим распределением и предполагаемым теоретическим распределением.

Эмпирические тесты

Приведенные эмпирические тесты применимы к последовательностям случайных чисел, равномерно распределенных между нулем и единицей.

Тест на монотонность (run test)

В данном тесте анализируются монотонные (возрастающие / убывающие) подпоследовательности в пределах одной сгенерированной последовательности случайных чисел. Рассмотрим последовательность из 10 цифр "1298536704" и разделим её на монотонные подпоследовательности вертикальными линиями

$$| 1 2 9 | 8 | 5 | 3 6 7 | 0 4 |.$$

В рассматриваемой десятичной последовательности выделено пять монотонных подпоследовательностей разной длины: две одночленные, одна двухчленная и две трехчленные:

$$(3) (1) (1) (3) (2).$$

В этом случае критерий χ^2 нельзя применять для анализа полученных данных, поскольку длины подпоследовательностей не являются независимыми (поскольку после длинной подпоследовательности чаще появляется короткая и наоборот). Вместо этого после определения длины подпоследовательностей вычисляется статистика:

$$V = \frac{1}{n} \sum_{i=1}^6 \sum_{j=1}^6 (B(i) - nb_i) \cdot (B(j) - nb_j) \cdot a_{ij},$$

где $B(k)$ – количество подпоследовательностей длины k ($k=1,2,3,4,5,\dots$), $B(6)$ – указывает количество подпоследовательностей длины 6 и более, a_{ij} , b_i – константы Кнута, коэффициенты критерия монотонности (приведены в табл. 19).

Чем больше значение V , тем ближе сгенерированная последовательность к теоретически независимой последовательности случайных чисел. Точно так же, как и во всех тестах, существуют допустимые значения минимального V , обеспечивающие определен-

ную степень надежности теста (например, для 95%-ной достоверности $V_{\min} = 12,6$, а для 99%-ной достоверности $V_{\min} = 16,8$ и т.д.). В зависимости от требуемой надежности рассчитанное V для последовательности сгенерированных случайных чисел сравнивается с предельным значением V_{\min} для этой степени надежности и делается вывод, является ли сгенерированная последовательность независимой по этому критерию.

Таблица 19

Коэффициенты критерия монотонности (константы Кнута)

$a_i =$	4529,4	9044,9	13568	18091	22615	27892
	9044,9	18097	27139	36187	45234	55789
	13568	27139	40721	54281	67582	83685
	18081	36187	54281	72414	90470	111580
	22615	45234	67852	90470	113262	139476
	27892	55789	83685	111580	139476	172860
$b_i =$	$\frac{1}{6}$	$\frac{5}{24}$	$\frac{11}{120}$	$\frac{19}{720}$	$\frac{29}{5040}$	$\frac{1}{840}$

Тест на равномерность (частотный тест)

Частотный тест используется для сравнения всего ряда случайных чисел с теоретическим равномерным распределением. Для этого интервал $(0, 1)$ разбивается на серию подынтервалов, и вычисляется суммарная относительная частота для верхней границы каждого из них. Если «-й» верхний предел равен x_i при соответствующей ему кумулятивной частоте F_i , так же, как и в случае с теоретической частотой $F(x) = x$, требуется абсолютная разность $F_i - x_i$, которая для каждого i должна быть ниже заданного максимума. Это можно сделать с помощью λ -теста. Для целочисленных последовательностей можно применить критерий χ^2 .

Серийный тест

Этот тест проверяет четность и независимость «-кортежей» (двойки, тройки, ...), образованных из последовательных чисел. Весь набор разбивается на «-кортежей» и определяется частота

встречаемости отдельных n -кортежей. Тест χ^2 сравнивает полученные частоты с теоретическими. (В этом случае теоретические частоты и вероятности равны $\frac{1}{d^n}$, где d – количество случайных чисел в наблюдаемой последовательности. Количество значений / классов равно d^n .)

Тест размаха (gap test)

В этом тесте проверяется длина интервала между появлением значений U_i , принадлежащих заданному интервалу. Если a и b две дроби, для которых справедливо $0 \leq a < b \leq 1$, то определим интервал (a, b) . Наблюдается последовательность равномерно распределенных чисел

$$U_{j-1}, U_j, \dots, U_{j+k-1}, U_{j+k},$$

из которых U_{j-1} и U_{j+k} попадают в интервал (a, b) . Эта последовательность определяет расстояние длины k . Так определяются частоты встречаемости отдельных пробелов. Если это

$$p = b - a = P[a \leq U_i \leq b],$$

тогда вероятность того, что расстояние равно k

$$p_k = p(1-p)^k.$$

Затем это распределение проверяется с помощью критерия χ^2 .

Покер-тест (проверка комбинаций)

В классическом покерном тесте рассматриваются n групп из пяти последовательных чисел: $Y_{5j}, Y_{5j+1}, \dots, Y_{5j+4}$ ($0 \leq j \leq n$). Возможны семь видов комбинаций:

- 1) abcde Все разные (1+1+1+1+1);
- 2) aabcd Одна пара (2+1+1+1);
- 3) aabbc Две пары (2+2+1);
- 4) aaabc Три одинаковых (3+1+1);
- 5) aaabb Полный (3+2);

6) aaaab Четыре одинаковых (4+1);

7) aaaaa Пять одинаковых (5).

Затем с помощью критерия χ^2 проверяется, соответствуют ли частоты появления комбинаций теоретическим вероятностям.

Перестановочный тест

Тест перестановки применяется к группам из n элементов в каждой. Весь интервал значений разбивается на «-интервалов», после чего определяются частоты для всех возможных перестановок n элементов. Все перестановки ($n!$ в каждой группе) равновероятны ($\frac{1}{n!}$), поэтому критерий χ^2 применяется к частотам различных перестановок.

Некоторые из представленных тестов относятся к статистическому анализу рядов за весь период. Однако в практических приложениях мы редко используем все случайные числа (весь период). Поэтому подходящими оказываются те тесты, с помощью которых можно проверить статистические характеристики отдельных участков (частей периода). Конечно, методы проверки рядов, интервалов, перестановок и т.д. мы можем успешно применить и к целым периодам.

Опыт тестирования случайных чисел показывает, что тест серии, тест частоты и тест перестановки являются «мягкими» и поэтому используются реже. Самые строгие проверки, т.е. те, которые приводят к отрицательному результату гораздо чаще, чем выпечеречисленные тесты, более сложные, и есть, прежде всего, тест на разрыв.

Каждый генератор случайных чисел служит определенной цели, поэтому можно сказать, что генератор случайных чисел хорош, если он хорош для конкретного приложения (редко он может быть хорош для всех случаев).

Глава 4. РАЗБОР ТЕКСТОВЫХ ИСТОЧНИКОВ ДЛЯ РЕКОНСТРУКЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ*

Большинство теорий –
лишь перевод старых мыслей
на новую терминологию.

Григорий Адольфович Ландау

4.1. Взаимосвязь смысла и структуры текстов

Гипотеза о том, что словоформы, их лексико-грамматическая дистрибуция и контекст, в рамках которого они реализованы, несут какой-то дополнительный смысл помимо основного денотативного и синтаксического значения, не вызывает сомнений. Мысль является стимулом к порождению какого-то смыслового единства, реализацией которого может стать некая языковая сущность от отдельного слова до целой комбинации языковых сочетаний, выстроенных в рамках определенной для каждого языка системы (форм словосочетаний, предложений подчинительного, сочинительного и др. типов синтаксической связи). Число таких реализаций не бесконечно, а конечно, имеет конечный набор структур, что и позволяет классифицировать все бесконечное множество языковых реализаций в виде ограниченного набора скелетных структур, на которых языковая система строится. Так мы наблюдаем взаимозависимую связь «человек – система», которая имеет способность существовать и дальше развиваться только при условии наличия каждой из составляющих этой связи (границы языка – границы мира).

Определение 4.1. *Скелетной структурой будем называть результат семантического и дистрибутивного разбора текста.*

Связь «человек – система» влечет за собой оппозицию «субъективное – объективное», которая может интерпретироваться как

* Глава написана А.В. Мыльниковой, Пермский национальный исследовательский политехнический университет.

внешнее – внутреннее, смысловое – структурное / скелетное. Эта связь неразрывна и проявляется в самом философском и лингвистическом понимании слова как двусторонней цельнооформленной и самостоятельно существующей лексико-грамматической единицы языка. Не вызывает сомнений тот факт, что семантический компонент слова имеет наибольший функциональный потенциал для выражения «мыслительной палитры» человеческой мысли, хотя известно, что и грамматические формы способны нести смысл и выражать определенную ситуацию: «Глѡкая кѹздра штѣко будланѹла бѡкра и курдѣчит бокрѣнка» (Л.В. Щерба). В данном случае речь идет о внутренней семантике предложения, выражающей взаимосвязи между участниками ситуации. Мы понимаем, что некий субъект женского рода совершает действие над другим субъектом мужского пола и затем продолжает делать что-то другое с более мелким представителем того же вида (возможно, его детенышем). При этом каждый может наложить свои собственные семантические признаки на актуализированные морфосинтаксические позиции: например, храбрая / черная собака / девочка быстро оттолкнула / повалила козла / кота и спасает / достает козленка / котенка. Данный пример наглядно показывает варианты реализаций внешней семантической языковой структуры в рамках скелетной структуры, приведенной в статье [77]. Для описания структур предложений системы английского языка вводятся такие базовые переменные, как NP1 (именная фраза), NP2 (объект), Aux (всегда изменяемый вспомогательный глагол у смыслового глагола), Gerund (герундий), V (инфинитив смыслового глагола) и др. Из этих переменных могут формироваться новые переменные или уточняться их значения для конкретизации типов предложений (по цели высказывания, наклонению, виду залога и т.д.).

Индивидуальность / уникальность человеческой мысли проявляется в игре со смыслами, которые появляются у слов внутри определенных контекстов, задаваемых автором, в нарушениях скелетных структур, придании словам не характерных для них синтаксических и грамматических значений, в создании новых слов и словосочетаний (неологизмов и т.д.). При этом не стоит забывать о том, что вся эта игра оказывается бессмысленной, если теряется

связь с онтологической базой языковой системы, с онтологическими законами человеческого мышления [78].

Среди группы концепций, в рамках которых предпринимались попытки истолковать лексическую семантику и ее внутреннюю структуру, распространены идеи, заложенные американскими учеными Дж. Остином и Ч.У. Моррисом, согласно которым выделяется три основных компонента реализации значения слова: прагматический, семантический и синтаксический. Дж. Остин и Дж. Серль выделяют систему речевых актов, которая основывается на положениях лингвистической прагматики [79, 80]. Теория дискурса Ю. Хабермаса представляет яркий пример современного развития теории коммуникативного действия в рамках междисциплинарного синтеза и сотрудничества философии, лингвистики, психологии и социологии – китов развития языковой системы [81, 82].

Наиболее плодотворным примером реализации языковой системы является поэзия, где слово наиболее ярко реализует в большей или меньшей степени все три выделенных выше компонента и является основным инструментом выражения истины, всех граней выражения правды [83] и призыва к действию, воздействия на человека / общество. К примеру, символистов российской поэзии А. Добролюбова, А. Блока, А. Белого, В. Брюсова и многих других объединяло недоверие к обыденному слову, стремление выражаться посредством аллегорий и символов. Символы играют важную организующую роль в художественном мире поэтов. По мнению А. Блока, «...осмыслить что бы то ни было вне символизма нельзя» [84]. В лирике А. Тарковского наряду с реализацией двоемирия «бытие – небытие» через группу символов, мы наблюдаем важный принцип «соответствия звука и цвета» [85], что дополнительно придает определенную эмоциональную тональность тексту. Постсимволисты (В. Ходасевич, О. Мандельштам, М. Цветаева, А. Ахматова, С. Есенин и др.), развивающиеся уже в среде многообразия творческих явлений и их «цветущей сложности», сформированной поэтами Серебряного века, пытаются найти свое слово и формируют «ансамблевую глоссолалию», прибавив, согласно словам критика Вяч. Иванова, к «вертикали» символизма (выражение человека

«в его свободном росте вглубь и ввысь») еще и «горизонталь» исторического бытия, психологического опыта реального проживания в истории [86], что в языкознании рассматривается с точки зрения теории пресуппозиций – необходимого семантического компонента слова, без которого нельзя адекватно понять текст. Лингвисты выделяют различные типы пресуппозиций в зависимости от аспекта их рассмотрения – когнитивно-прагматического, культурно-языкового, логического, но некоторые классификации используют целостный подход к классификации. К примеру, Н.Д. Арутюнова выделяет такие виды пресуппозиций, как экзистенциальные, прагматические, синтагматические, коннотативные, коммуникативные и логические [87, 88].

Наряду с усложнением и все большей изощренностью выражения семантического и прагматического компонентов слова, наблюдается также тенденция достижения «инового» смысла за счет обращения к внутренней структуре языка, к его синтаксической компоненте. К примеру, В. Маяковский достигает прагматического, воздействующего эффекта за счет вольного обращения с синтаксическими структурами, создавая феномен рваных строк (*bizarre inventions*) [89].

Существует мнение, что смысл кроется не в словах, а в связи слов. Примером этого является следующий текст (хотя ни одно слово изначально неизвестно, но после прочтения смысл становится очевидным) [90]:

«Сяпала Калуша с калушатами по напушке. И уварила бутявку и волит: (– Калушата, Калушатки! Бутява!) Калушата присяпали и бутявку стрямкали. И подудонились. А Калуша волит: (– Оеё! Оеё! Бутявка-то некузявая!) Калушата бутявку вычучили. Бутявка вздрезднулась, сопритюкнулась и усяпала с напушки. А Калуша волит калушатам: (– Не тремкайте бутявок, бутявки дыбые и зюмо-зюмо некузявые. От бутявок дудонятся.) А бутявка волит за напушкой: (– Калушата подудонились! Калушата подудонились! Зюмо некузявые! Туськи бетые!».)

Данная тенденция получила развитие в создании компьютерных стихов, но дальше развиваться не стала, так как результаты

машины не способны «продолжать жить» в языковом пространстве социума и поэтому быстро забываются и представляют интерес только с точки зрения экспериментальных данных.

Вышеприведенный анализ позволяет сделать вывод, что обращение ученых к тексту как к источнику смысла существует так долго, как существует сам язык, и имеет разную направленность и разные задачи.

В настоящее время объем информации, с которым мы имеем дело, стремительно растет. В связи с этим актуальной становится задача обработки данных. В том числе и текстовых. В качестве примера задач можно рассмотреть определение образа о некотором объекте или субъекте (например, политике или фирме) или, наоборот, его формирование через заметки в СМИ, определение отношения отдельных субъектов к отдельным вопросам по его высказываниям и т.п.

Кроме того, развитие информационных технологий ставит перед нами такие задачи, как выявление агрессивных комментариев или рекламы под сообщениями в группах и чатах.

Важными характеристиками текста являются его тип (статья в газете (публицистика), художественный текст, юридический документ, инструкция к техническому устройству, молитва и т.д.), общая тональность, наличие в нем возможных миров (попытка расширить оппозицию «границы языка – границы мира» через создание множества разных миров за счет способности языковой сущности развиваться и приспосабливаться к новым контекстам своего существования, устанавливаемых создателем – человеком: например, искусственные языки вымышленных вселенных Дж.Р.Р. Толкина, клингонский язык из сериала «Звездный путь» (Star Track), язык эсперанто; субъязыки сообществ геймеров, программистов и другие).

Решение задач анализа текстов прошло путь от сбора статистики и вычисления специфических характеристик (индекс дистрибуции, индекс итерации, индекс исключительности, индекс предсказуемости, индекс плотности текста, индекс стереотипности и т.д.)

до использования методов машинного обучения для выявления специфических характеристик текстовой информации. Использование методов машинного обучения, как правило, требует предварительной подготовки данных. Методы работы с текстами, которые при этом применяются, можно разделить на две группы:

1) методы, работающие с отдельными символами, составляющими текст:

– определение тональности текста в области анализа мнений (к примеру, особенно актуальным является анализ тональности новостных текстов, так как обнаружение, отслеживание трендов тональности в новостном потоке важно для построения разного рода аналитических систем, отслеживания имиджа в СМИ конкретных людей или компаний);

– анализ звукоцвета текста (фоносемантика) с целью определения эмоциональной окраски текста (является дополнительным инструментом для определения тональности текста);

2) методы, рассматривающие слова или словоформы как отдельные единицы (как правило, получающиеся после токенизации текста, удаления «стоп»-слов и специальных символов, лемматизации или стемминга):

– Bag of Words (в этой модели текст (одно предложение или весь документ) представляется в виде множества его слов без какого-либо учета грамматики и порядка слов, но с сохранением информации об их количестве);

– TF-IDF в дополнении к предыдущему подходу использует статистическую меру, используемую для оценки важности слова в контексте документа;

а также Word2Vec [91], fastText с использованием n -грамм и др.

Как следует из исследований, большое количество значимой информации скрыто в самой структуре текста и во взаимосвязях используемых слов. Этот фактор будет иметь особенно большое значение для языков со слабо фиксированной языковой структурой. Однако в настоящее время для анализа текстов данные о структуре языка в явном виде не используются.

Обнаружение тональности текста возможно и за счет обращения к фоносемантике, согласно которой слово обладает двумя свойствами – имеет понятийное ядро и ореол. Понятийное ядро заключает в себе смысл, ореол – окраску слова. Например, слова «папа», «папулечка», «отец» несут один смысл (родитель – мужчина), но разную эмоциональную окраску. Но если ореол объективно существует, то его можно измерить. При этом смысл всей фразы и эмоциональный окрас может скрываться и в структуре предложения или части текста.

Для того, чтобы иметь возможность учитывать структуру языка, необходим метод, позволяющий её представить в доступном для алгоритмической обработки виде.

При разработке набора данных необходимо выделить крупные группы, описывающие языковые структуры: 1) морфология (самостоятельные и служебные части речи), 2) синтаксис (сочинительные и подчинительные союзы) и 3) пунктуация (точка, запятая, двоеточие и т.д.). Первая группа, безусловно, более разнообразна и ввиду специфики выбранного материала имеет разнообразие элементов, влияющих на тональность текста, его эмоциональный окрас, а именно целый класс частиц, наречий и местоимений. Особенностью построения скелетных структур является не только возможность учета самого факта наличия того или иного элемента, но и место его реализации как в отдельной синтаксической единице (предложении), его окружении, так и в рамках / границах всего исследуемого текста.

Факультативные элементы (модальные слова, частицы, наречия, междометия) служат для актуализации субъективно-окрашенного семантического окраса ситуации (соответствуют эпистемическому и аксиологическому семантическим уровням отношения субъекта к актуализируемой ситуации), который накладывается на базовый онтологический семантический уровень (рис. 42) [78].

Использование предложенной разметки опирается на предположение о том, что каждый из текстов имеет структурные особенности.



Рис. 42. Модель взаимодействия семантических уровней, актуализируемых скелетной языковой структурой

Разметка позволяет получать не только закодированный текст, описанный в виде набора базовых элементов, но и статистические данные, такие как номера позиций, занимаемых базовыми элементами в тексте. При статистической обработке таких данных мы можем получить важную информацию о частотности реализации того или иного элемента, о наличии или отсутствии значимых дистрибуций, характерных для того или иного типа текстов (см. табл. 20).

Таблица 20

Пример разметки четверостишья С. Есенина

Позиция	1	2	3	4	5	6	7	8
Обозначение	@N	@Dash	@N	@N	@Comma	@Part.neg	@Vf	@Prep
Фраза	Упоенье	–	яд	отравы	,	Не	живи	среди

9	10	11	12	13	14	15	16	17	18	19
@N	@Comma	@Part.neg	@Vf	@Pron.pos	@N	@Prep	@N	@Adj	@N	@Dot
людей	,	Не	меняй	своей	забавы	На	красу	бесцветных	дней	.

Такое преобразование позволяет формализовать представление языковой структуры и выдвинуть гипотезу о том, что это позволит находить различия / особенности в построении структур предложений и при рассмотрении больших объемов текста определять тип текста и/или автора текста.

***Определение 4.2.** Дескриптором будем называть основную смысловую часть предложения, которая может быть описана элементами из табл. 20.*

***Определение 4.3.** Спецификатором будем называть семантически-дистрибутивные особенности построения фраз, включающие семантические единицы, не входящие в дескриптор.*

***Предположение 4.1.** Скелетная структура состоит из дескриптора и спецификатора, каждый из которых может использоваться для выявления особенностей текстов.*

В этом случае дескриптор [92] можно рассматривать как структуру «субъект – действие – объект – окружение», описанную в [70] и использующуюся для описания структурно-функциональных и алгоритмических моделей. В этом смысле выделение дескриптора позволяет перейти к автоматизированному построению функциональных схем на основе текстовых данных, формирования поисковых запросов и составления рефератов на основе авторских текстов.

Спецификатор, в отличие от дескриптора, в этом случае будет содержать либо шумовую составляющую текста, либо его специфические особенности. Такое разделение на дескрипторы и спецификаторы позволяет говорить о возможности решения классических задач анализа текстов, связанных с определением авторства, стиля изложения и др.

4.2. Формализация структуры текстов на основе скелетных структур

Для описания структуры текстов на основе скелетных структур был создан язык описания, для каждого события была построена классификация наиболее частотных форм его реализации (набора ситуаций на основе деловой коммуникации), что и позволило выявить скелетные языковые структуры, на которые опирается полученный язык и описания.

Для описания языковых структур использовались следующие обозначения:

- () – определение типа предложения;
- = – раскрытие понятия;
- { } – объединение;
- <> – обязательный член предложения;
- [] – необязательный член предложения;
- | – или;
- "" – жестко заданный текст;
- \ – определение новой переменной.

Тогда описание текста состоит из двух составляющих: описание типа предложения и описание структуры предложения.

Для описания типов предложений введем следующие классификаторы и обозначения:

1. Классификация по цели высказывания: повествовательное (1А), вопросительное (1В), побудительное (1С).

2. Классификация по форме наклонения: изъявительное (2А), условное (2В), повелительное (2С).

3. Классификация по виду залога: активный залог (3А), пассивный залог (3В).

4. Классификация по виду временной группы: прошедшее простое время (4А), настоящее простое время (4В), настоящее в процессе (4С), прошедшее в процессе (4D), настоящее с результатом (4Е), прошедшее с результатом (4F), настоящее с результатом и процессом (4G), прошедшее с результатом и процессом (4I), будущее простое (4J), будущее в прошедшем (4Н), будущее в прошедшем с результатом (4К), близкое будущее (4L), близкое будущее в прошлом (4М), незамедлительное будущее (4N), незамедлительное будущее в прошлом (4Р).

5. Классификация по характеру связи между предметом речи и тем, что о нем сообщается: утверждение (5А), отрицание (5В).

6. Классификация по форме сказуемого: простое сказуемое (выраженное смысловым глаголом или связочным глаголом) (6А), сказуемое с модальной оценкой (модальные глаголы) (6В), сказуемое с аспектной оценкой (аспектные глаголы) (6С), сказуемое с модальной и аспектной оценкой (модальный +аспектные глаголы)

(6D), сказуемое со смысловой оценкой (глаголы со значением предположения, предложения, желания, рекомендации, совета) (6E), сказуемое с модальной и смысловой оценкой (модальный глагол+ глагол со значением предположения, предложения, желания, рекомендации, совета) (6F).

7. Могут вводиться дополнительные типы классификаций в зависимости от языка (например, для английского языка может быть введена классификация по словам, меняющим структуру предложений).

Для описания структуры предложений необходимо введение таких базовых переменных, как именная фраза (подлежащее) (NP1), объект (дополнение) (NP2), вспомогательный глагол у смыслового глагола (всегда изменяемый) (Aux), герундий (Gerund), инфинитив (смыслового глагола) (V), дополнение (NP3), прилагательное (Adj), определительное (Det), определенный артикль (Art/def), неопределенный артикль (Art/indef) и др. (см. листинг 1).

Листинг 1. Описание базовых переменных.

```
@NP1 <- {"Именная фраза (подлежащее): кто/что делает?"}
```

```
@NP2 <- {"Объект (дополнение): Делает что?"}
```

```
@NP3 <- {"Объект (дополнение): Делает где? (например, в компании). Не используется вместе с AdvP3!)"}
```

```
#@NP2ext <- {"Раскрытие/ описание объекта: Кого/чего? Для кого/для чего? Кому/ чему? Кем/Чем? С кем/чем? ..."}
```

```
@Aux <- {"Вспомогательный глагол у смыслового глагола (всегда изменяемый)"}
```

```
@AuxE <- {"Вспомогательный глагол у настоящего времени с результатом have (всегда изменяемый)"}
```

```
@AuxE2 <- {"Вспомогательный глагол у прошедшего времени с результатом had (неизменяемый)"}
```

```
@Gerund <- {"Герундий"}
```

```
@V <- {"Инфинитив (смыслового глагола)(to do)"}
```

```
@V1.1.1 <- {"Инфинитив связочного глагола (to be)"}
```

```
@V1 <- {"1-я форма смыслового глагола (do)"}
```

```
@V1.1 <- {"1-я форма связочного глагола (be)"}
```

```
@V2 <- {"2-я форма смыслового глагола (did)"}
```

@V2.1eg. <- {"2-я форма связочного глагола ед.ч. (was)}
 @V2.1pl. <- {"2-я форма связочного глагола мн.ч. (were)}
 @V3 <- {"Например, выполненный, составленный"}
 @V3.1 <- {"3-я форма связочного глагола (been)}
 #@Vf1 <- {"Изменяемая часть смыслового глагола (makes)}
 @Vf1 <- {"Глагол (делаю, смотрю...)}
 @Vf1.1 <- {"Изменяемая часть связочного глагола (is...)}
 @Vg <- {"Герундий, образованный от смыслового глагола (making)}
 @V1g <- {"Герундий, образованный от связочного глагола (being)}
 #@VPvf1 <- {"Предикат, выраженный смысловым изменяемым глаголом (makes)}
 @VPvf1 <- {"Глагол (делаю, смотрю...)}
 @VPvf1.1 <- {"Предикат, выраженный связочным изменяемым глаголом (is...)}
 @VPvf1.2 <- {"Описание предиката с помощью именной части (likes doing/prepares to do)}
 @VPvflvg <- {"Предикат с именной частью NVPvg (likes doing)}
 @VPvflv1g <- {"Предикат с именной частью NVPv1g (likes being)}
 @VPvflNVPv <- {"Предикат с именной частью NVPv (prepares to do)}
 @VPvflNVPv1.1.1 <- {"Предикат с именной частью NVPv1.1.1 (prepares to be)} (для внутреннего анализа = если be, то ограничивается набор репрезентантов объектов)
 @VPm <- {"Описание предиката с помощью глаголов модального значения (can do)}
 далее в зависимости от выбора предиката система выбирает один из ниже приведенных вариантов:
 @VPm1 <- {"Предикат, выраженный модальным глаголом с именной частью NVPv1(can do)}=<"can| could| must| shall| should| may| might| would" V1|V1.1.>}

@VPm2 <- {"Предикат, выраженный модальным глаголом или смысловым глаголом модального значения с именной частью (ought to do/needs to do)}=<"want | need| have| ought| be| dare| used| be able| be allowed| " V | V1.1.1>}

@VP1auxnotB <- {"Изменяемая часть вспомогательного глагола в настоящем простом времени с отрицанием (полная форма)"}

@VP1auxB <- {"Изменяемая часть вспомогательного глагола в настоящем простом времени в вопросительном предложении (полная форма)"}

@VPAuxE <- {"Изменяемая часть вспомогательного глагола в настоящем времени с результатом (полная форма – have)"}

@VPAuxE2 <- {"Неизменяемая часть вспомогательного глагола в прошедшем времени с результатом (полная форма – had)"}

#@VP3<- {"Третья форма смыслового глагола – неизменяемая (done)"}

@VP3<- {"Глагол (сделал, посмотрел)"}

@“not” <- {"Неизменяемая частица при отрицании"}

@VP1auxntB <- {"Изменяемая часть вспомогательного глагола в настоящем простом времени с отрицанием и в вопросительном предложении (краткая форма)"}

@VPvf1.1ntB <- {"Изменяемая часть связочного глагола в настоящем простом времени с отрицанием в вопросительном предложении (краткая форма)"}

@VP2v1 <- {"Вторая часть предиката при отрицании и вопросе (смысловой глагол)"}

@VP2v1vg <- {"Вторая часть предиката при отрицании и вопросе (смысловой глагол) с именной частью NVPvg (like doing)"}

@VP2v1NVPv <- {"Вторая часть предиката при отрицании и вопросе (смысловой глагол) с именной частью NVPv (make to do)"}

@VP2v1NVPv 1.1.1 <- {"Вторая часть предиката при отрицании и вопросе (смысловой глагол) с именной частью NVPv1.1.1 (make to be)"}

#@AdvP1 <- {"Адвербиальная фраза (наречия неопределенного времени)"}

#@AdvP2 <- {"Адвербиальная фраза (наречия определенного времени, места; именные фразы в функции обстоятельства)"}

#@AdvP2ext <- {"Расширение адвербиальной фразы с помощью наречия определенного времени, места или именные фразы в функции обстоятельства"}

#@AdvP1 <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: this time, this year, today, yesterday, last month)"}

@AdvP1 <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: в этот раз, в этом году, сегодня)"}

@AdvP2 <- {"Уточнение предиката: каким образом делает? Не используется вместе с AdvP1.1, AdvPM, Adv/measure"}

@AdvP3 <- {"Уточнение предиката: Где делает? Не используется вместе с NP3"}

@AdvP11 <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: this time, this year, today, yesterday, last month)"}

#@AdvP1/1 <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: this time, these days, now, at the moment, currently, today, nowadays, presently"}

@AdvP1/1 <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: в этот раз, в эти дни, сейчас, в данный момент ..."}

#@AdvP1/1.1<- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: this time, now, at the moment, currently, today, presently)"}

#@AdvP1/1x <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: right now"}

#@AdvP1x/1x <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: right now"}

@AdvP1/2<- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: that time, last month, yesterday, in 1998, in the 50s, once, upon a time)"}

@AdvP1/2x <- {"Уточнение предиката: когда точно делает? (наречия определенного временного периода: ago")}

@AdvP1/3 <- {"Уточнение предиката: когда точно делает? ("наречия определенного временного периода: next time, in two days, in future, tomorrow")}

@AdvP1.1 <- {"Уточнение предиката: как регулярно делает? (наречия определенной частоты) Не используется вместе с AdvP2 ! (ежедневно, ежегодно ...)"}

@AdvP1.1x <- {"Уточнение предиката: как регулярно делает? (наречия определенной частоты: daily, annually, every day/hour/month/year, once a week/month/year etc)"}

@AdvP1.2 <- {"Уточнение предиката: как часто делает? (наречия неопределенной частоты: всегда, часто, редко ...)"}

@AdvP1.3 <- {"Present Continuous (если нет маркеров Past Simple), если есть, то Past Continuous"}

@AdvP1.3/1 <- {"Present Perfect (если нет маркеров Past Simple, если да, то Past Perfect)"}

@AdvP1.3/1x <- {"Present Perfect (если нет маркеров Past Simple, если да, то Past Perfect)"}

@AdvP1.3/1.1 <- {"Используется только с AdvP1.3/5 = "еще"..."}

@AdvP1.3x/1.1 <- {"Не используется вместе с AdvP1.3/5, NP3 ! Например, до сих пор..."}

@AdvP1.3/2 <- {"К настоящему моменту, раньше, в последнее время ..."}

@AdvP1.3x/2 <- {"Не используется вместе с NP3 ! "С 1985, за последние две недели, в течение прошлого месяца ..."}

@AdvP1.3/2x <- {"раньше", " в последнее время" ...}

@AdvP1.3/3 <- {"Past Perfect"}

#@AdvP1.3/4 <- {"ever"}

@AdvP1.3/4 <- {"Используется только с AdvP1.3/2x = "раньше"}

@AdvP1.3/5 <- {"Не используется вместе с AdvP1.3/2 ! Например, еще, сто лет, уже сто лет, раньше, в последнее время ..."}

@Vaspf1 <- {"Изменяемый аспектный глагол, выражающий значение начала действия"}

@Vaspf2 <- {"Изменяемый аспектный глагол, выражающий значение продолжения действия"}

@Vaspf3 <- {"Изменяемый аспектный глагол, выражающий значение окончания действия"}

@Det <- {"Определительное"}

@Art/def <- {"Определенный артикль"}

@Art/indef <- {"Неопределенный артикль"}

@- <- {"Нулевое (пустое) место"}

@Adj <- {"Прилагательное; например, современный, сложный, программный ..."}

@Pron/pos <- {"Притяжательное местоимение; например, мой, наш, твой ..."}

@Pron/det <- {"Указательное местоимение; например, этот, тот ..."}

@Pron/pers <- {"Личное местоимение; например, я, мы, они ..."}

@Pron/attr <- {"Определительное местоимение (включая отрицательные местоимения); например, все, всё, никакой ..."}

@Adv/indef.time <- {"Наречие неопределенного времени"}

@Adv/def.time <- {"Наречие определенного времени"}

@Adv/place <- {"Наречие места"}

@Adv/measure <- {"Наречие меры не используется вместе с AdvP1.2, AdvP1.3., AdvPM! Например, очень, сердечно, действительно ..."}

@Adv/action desc. <- {"Наречие образа действия"}

@AdvPM <- {"Наречие с модальным окрасом не используется вместе с AdvP1.2, AdvP1.3; например, скорее всего, вероятно, вряд ли ..."}

@AdvPMod.M.0.9 <- {"Modal modifier/ наречие с модификатором высокой степени предположения"}

@AdvPMod.M.0.5 <- {"Наречие с модальным модификатором средней степени предположения"}

@AdvPMod.M.0.1 <- {"Наречие с модальным модификатором низкой степени предположения"}

@Vmod.M.0.9 <- {"Verbal modal modifier / смысловой глагол с модальным модификатором высокой степени предположения"}

@Vmod.M.0.5 <- {"Смысловой глагол с модальным модификатором средней степени предположения"}

@Vmod.M.0.1 <- {"Смысловой глагол с модальным модификатором низкой степени предположения"}

@Comma <- {"Запятая"}

@Conj. <- {"conjunction/союз"}

@Sub.expl.1 <- {"subordinate explanatory / подчиненный изъяснительный (that)"}

@Sub.expl.2 <- {"subordinate explanatory / подчиненный изъяснительный (in order to)"}

@Sub.cond.1 <- {"subordinate conditional / подчиненный условный (if)"}

@Sub.cond.2 <- {"subordinate conditional / подчиненный условный (when)"}

@Sub.cond.3 <- {"subordinate conditional / подчиненный условный (whenever)"}

@VPdes1 <- {"VP desire 1/ сказуемое, выражающее значение желания (would like)"}

@VPdes2.Vfl <- {"VP desire 2/ сказуемое, выражающее значение желания (wish) в изменяемой форме"}

@VPdes3.Vfl <- {"Сказуемое, выражающее значение желания (hope) в изменяемой форме"}

@ImpS <- {"impersonal subject/ безличное подлежащее (it)"}

@Adj.qualit. <- {"qualitative adjective / качественное прилагательное (good, excellent)"}

@Adj.quantit. <- {"quantitative adjective / относительное прилагательное (sunny, wintry, golden)"}

@VPprop.1 <- {"Сказуемое, выражающее значение предложения (let) в неизменяемой форме (повелительное наклонение)"}

@VPprop2 <- {"Сказуемое, выражающее значение предложения (would be) в неизменяемой форме"}

@VPprop3.Vfl <- {"Сказуемое, выражающее значение предложения (suggest) в изменяемой форме"}

@VPprop4.Vfl <- {"Сказуемое, выражающее значение предложения (offer) в изменяемой форме"}

@NP1prop <- {"<Det><N>=<Pron/pos><suggestion>"}
 @VPrecom1.Vf1 <- {"Сказуемое, выражающее значение рекомендации (recommend) в изменяемой форме"}
 @VPadv.1.Vf1 <- {"Сказуемое, выражающее значение совета (advise) в изменяемой форме"}
 @Interrog.word <- {"Вопросительное слово"}
 @Interrog.word/cause <- {"Вопросительное слово причины (why)"}
 @N <- {"Существительное или словосочетание (например, портфель или портфель проектов)"}
 @Prep <- {"Предлог; например, в, на, по ..."}
 @Adv <- {"Наречие"}
 @NP2v1 <- {"Именная фраза после изменяемого смыслового глагола (make)"}
 @VP2v1v1g <- {"Вторая часть сказуемого при отрицании и вопросе (смысловой глагол) с именной частью (like being)"}
 @VP2v1NVPv1.1.1 <- {"Вторая часть сказуемого при отрицании и вопросе (смысловой глагол) с именной частью (make to be)"}
 #@NP3 <- {"Дополнение"}
 #@VPf1.1 <- {"Предикат, выраженный связочным изменяемым глаголом (is...)"}
 @VPf1.1 <- {"Для русского языка не используется"}
 #@VPg <- {"Герундий, образованный от смыслового глагола (making)"}
 @VPg <- {"Глагол (делаю, смотрю...)"}
 @NP2ext <- {"Расширение дополнительной именной фразы с помощью дополнительной именной фразы с предлогом: Кого/чего? Для кого/для чего? Кому/ чему? Кем/Чем? С кем/чем?"}
 @AdvP1.3x1.1 <- {"so far,"}
 @VP <- {"Что делает? Что сделал? (в зависимости от выбранного времени)"}
 @не <- {"Отрицательная частица – не"}
 #@Parenth <- {"Parenthesis; for example, alternatively, in other words ..."}
 @Parenth <- {"Вводные слова; например, в качестве альтернативы, другими словами ..."}

Из этих переменных могут формироваться новые или уточняться их значения (см. листинг 2) для конкретизации типов предложений (см. листинг 3).

Листинг 2. Пример описания новых переменных через базовые.

```

\AdvP1 = [AdvP1.2][AdvP1.1x]
\NP2v={<Det><N>}|{|<Prep><Det><N>}|{|<Pron/pers>}|{|<Adv><Adj>}
\NP2v1g={<Adj>}|{|<Det><N>}|{|<Prep><Det><N>}|{|<Adv><Adj>}|{|<Pron/pos>}
\AdvP2 = {"slowly"}|{"quickly"}|{"rapidly"}
\AdvP3      =      {"here"}|{"there"}|{"outside"}|{"inside"}|{"somewhere"}|{"everywhere"}

```

Листинг 3. Пример описания универсальных структур предложений.

```

(1A,2A,3A,4B,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1][AdvP2][AdvP3]<". ">
(1A,2A,3A,4B,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1][AdvP2][AdvP3]<". ">
(1B,2A,3A,4B,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1][AdvP2][AdvP3]<"?">
(1B,2A,3A,4B,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1][AdvP2][AdvP3]<"?">
(1A,2A,3A,4C,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]<". ">
(1A,2A,3A,4C,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]<". ">
(1B,2A,3A,4C,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]<"?">
(1B,2A,3A,4C,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]<"?">
(1A,2A,3A,4E,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]<". ">

```

(1A,2A,3A,4E,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]
]<".">

(1B,2A,3A,4E,5A,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]
]<"?">

(1B,2A,3A,4E,5B,6A)=<NP1><VP><NP2>[NP2ext][NP3][AdvP1]
]<"?">

Формальное описание позволяет выделять конструкции и для установления соответствий со структурно-функциональным представлением.

Такой вид описания позволяет также создавать конструкторы фраз, которые в силу ограничений, искусственно вносимых разработчиками в конструкции фраз, будут выступать входным фильтром и инструментом для подготовки текстовых описаний структурно-функциональных моделей.

4.3. Некоторые аспекты реконструкции информационных процессов из текстов

Реконструкция текстов имеет смысл, если они содержат описание последовательности некоторых действий / алгоритм / инструкцию. К таким текстам можно отнести инструкции пользователя, рецепты.

Рассмотрим рецепт классической шарлотки (см. источник: <https://eda.ru/recepty/vypechka-deserty/klassicheskaja-sharlotka-21916>):

ИНГРЕДИЕНТЫ

Сахар – 1 стакан

Куриное яйцо – 5 штук

Пшеничная мука – 1 стакан

Яблоко – 7 штук

Растительное масло – 1 столовая ложка

Разрыхлитель – ½ чайные ложки

Шаг 1. Разогреть духовку. Отделить белки от желтков. Белки взбить в крепкую пену с щепоткой соли, постепенно добавляя сахар. Инструмент – термометр для духовки.

Шаг 2. Продолжать взбивать, добавляя по одному желтки, затем разрыхлитель и муку. Тесто по консистенции должно напоминать сметану. Инструмент – сеелка для муки.

Шаг 3. Смазать противень растительным маслом. Вылить половину теста на противень, разложить равномерно нарезанные дольками яблоки, залить второй половиной теста.

Шаг 4. Поместить противень в разогретую духовку. 3 минуты подержать при температуре 200 градусов, затем убавить до 180 и выпекать 20–25 минут.

Такого рода описание может быть представлено схемой, приведенной на рис. 43.



Рис. 43. Структурно-функциональная модель приготовления классической шарлотки

Рассмотрим другой пример – часть инструкции по пользованию утюгом (см. источник: https://cdn.rbt.ru/images/itm/file_manual/526/525293_312911607_195d3e88269ba7268b82c8986a4e4725.pdf):

*Установите регулятор температуры в положение *tip* и отсоедините утюг от электросети. Установите утюг в вертикальное положение и дайте ему полностью остыть. Слейте воду из емкости утюга. Удалите отложения с подошвы утюга с помощью раствора уксусной кислоты. Протрите корпус утюга влажной тканью, затем вытрите насухо.*

Такое описание также может быть представлено структурно-функциональной схемой (рис. 44).

Как видно из приведенных примеров, реконструкция может выглядеть по-разному. Несмотря на различия, при реконструкции её основой являются действия, условия, в которых они реализуются (окружение), и результаты, к которым они приводят. Если уда-

ется выделить субъекта управления, действие (может быть сложным), объект действия, то такая реконструкция оказывается возможной.

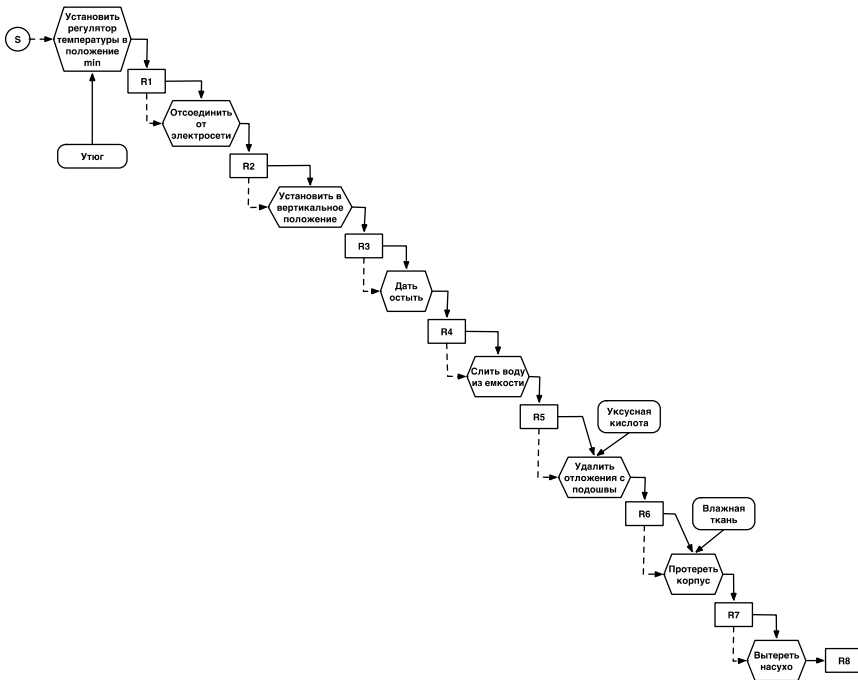


Рис. 44. Структурно-функциональная модель чистки подошвы утюга

Продemonстрированная разница основана на разнице текстов. Таким образом, для каждого типа текста необходимым становится описание своей структуры, как приведено выше. Тогда становится актуальной задача идентификации типов текстов для разбора этих структур.

Эксперимент 1. Определение популярности текста

Если предположить, что при анализе структуры текста существует такое количество просмотров, после превышения которого однозначно будет определяться, к какой категории (больших или низких просмотров / качества) относится текст, тогда можно прове-

сти эксперимент, разметив данные таким образом, чтобы разделить их на два класса, означающих большое и малое количество просмотров.

Для проведения эксперимента возьмем данные с сайта Kaggle (<https://www.kaggle.com/datasets/thegupta/ted-talk>), которые содержат текст выступлений и информацию о количестве просмотров. Стенограммы выступлений подготовлены на английском языке. Данные содержат 4609 текстов выступлений с количеством просмотров от 0 до 65 678 748 и медианном значении длины транскрипции выступления в символах, равным 8819. Для анализа будем определять части речи и составим таблицу для выступлений, в которой для каждого вычисления будет приведено среднее местоположение частей речи.

На основании этих данных будем решать задачу бинарной классификации. Использование любых методов машинного обучения не меняет качественную картину точности их определения (рис. 45).

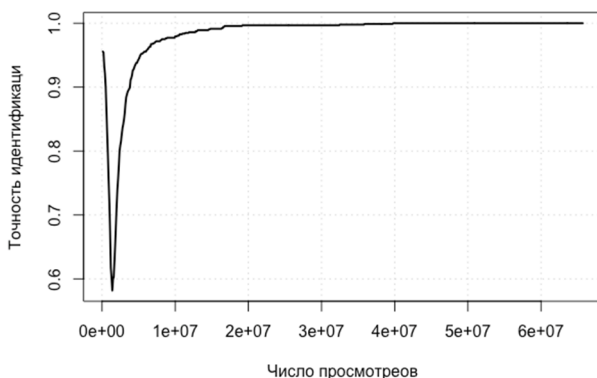


Рис. 45. График значения точности (ассурагу) определения класса, к которому относится текст при изменении границы разделения по числу просмотров текстов на классы популярных и непопулярных методом k NN

Из проведенного эксперимента видно, что при изменении границы, по которой будет определяться, к какой категории будет относиться текст, существует некоторая область смещения, в которой не удастся разделить тексты на категории.

Эксперимент 2. Определение категории текста

Для проведения эксперимента возьмем новостные тексты, разбитые по тематическим категориям, приведенным на сайте Kaggle (<https://www.kaggle.com/datasets/qusaybtoush1990/english-news>). Выборка содержит 1490 записей на английском языке, для каждой категории от 250 до 350 значений при медианном значении длины новостной заметки в символах, равном 1961.

Для определения будем выбирать одну из категорий (business, sport, entertainment, politics, tech). Таким образом, сведем задачу к задаче бинарной классификации, присваивая выбранной категории значения класса =1, а остальным категориям – значения класса = 0, чтобы данные были сбалансированы. В результате получаем результаты, приведенные в табл. 21.

Таблица 21

Точность (Accuracy) работы классификаторов при выделении выбранной категории из всего множества новостных текстов

kNN	SVM	C5.0	Naïve Bayes	gbm
<i>Категория tech соответствует классу =1</i>				
0.67	0,78	0,82	0,74	0,82
<i>Категория politics соответствует классу =1</i>				
0.59	0,70	0,74	0,71	0,75
<i>Категория entertainment соответствует классу =1</i>				
0.62	0,72	0,76	0,58	0,71
<i>Категория sport соответствует классу =1</i>				
0.64	0,80	0,79	0,77	0,81
<i>Категория business соответствует классу =1</i>				
0.56	0,76	0,80	0,67	0,76

Несмотря на то, что разные методы показывают разную точность для разных категорий, видим, что получаемые результаты неслучайны. Таким образом, можно говорить, что категорию текста получается определить на основе предлагаемого подхода.

Дополнительные эксперименты

В литературе описан успешный способ определения авторства с использованием описанного подхода (см., например, [93]) на примере полных стихов А. Блока и С. Есенина на языке оригинала (русский язык).

Определение эмоционального окраска

Рассмотрим данные, представленные для анализа на конкурсе RuSentNE-2023 (<https://codalab.lisn.upsaclay.fr/competitions/9538#results>), проводимым совместно МГУ и Университетом Ньюкасла. Набор данных содержит 6637 записей, размеченных как -1 – негативный окрас, 0 – без эмоционального окраса и +1 – позитивный окрас. Медианное значение длины текста в символах равно 138. Тексты составлены на русском языке.

В результате анализа группы текстов на специально подготовленных сбалансированных выборках видно, что эмоциональный окрас определить не удастся. Получаемые результаты близки к 50 %, что говорит о случайном характере получаемых ответов при рассмотрении классов по отдельности. При использовании несбалансированных выборок модели склонны ставить всей или почти всей выборке признак одного из классов (табл. 22, 23).

Таблица 22

Точность (Ассигасу) работы классификаторов при выделении выражений с позитивным окрасом в класс 1, а остальных выражений в класс 0

kNN		SVM		C5.0		Naïve Bayes		gbm	
Класс		Класс		Класс		Класс		Класс	
1	0	1	0	1	0	1	0	1	0
0,583	0,533	0,577	0,568	0,565	0,590	0,005	0,989	0,607	0,556

Таблица 23

Точность (Ассигасу) работы классификаторов при выделении выражений с негативным окрасом в класс 1, а остальных выражений в класс 0

kNN		SVM		C5.0		Naïve Bayes		gbm	
Класс		Класс		Класс		Класс		Класс	
1	0	1	0	1	0	1	0	1	0
0,562	0,536	0,597	0,508	0,706	0,400	0,547	0,571	0,547	0,584

Одной из причин таких результатов может являться то, что подобранные выражения состоят из одного предложения (заголовков новостных лент), и их понимание без контекста текущей ситуации затруднено. Это отражает сложность решения такой задачи в общем виде.

Определение региона

Еще одной задачей классификации, которая позволяет сузить область поиска, является задача определения региона, к которому относится текст новости, опубликованной в газете. Если провести эксперимент, аналогичный предыдущим, на основе данных с сайта: <http://news.pfo-perm.ru/>, собранных за два месяца (апрель и май 2023 года), то получаем результат, не позволяющий сделать выводы о принадлежности к заданному региону (точность $\sim 0,5$). Точность работы моделей повышается до устойчивых 60 % (т.е. становится не случайной), если рассматривать макрорегионы (например, Уральский регион как состоящий из Пермского края, Удмуртской Республики, Тюменской, Свердловской, Кировской и Челябинской областей). Рассматриваемое число записей составило 5191 при медианном значении длины новости, равном 1627 символов. Тексты составлены на русском языке.

Такие результаты оказываются недостаточными для решения задачи фильтрации текстов по региону в системах поиска.

Определение категории текста

Для проведения эксперимента возьмем новостные тексты, взятые с сайта BBC и разбитые по тематическим категориям, которые приведены на сайте Kaggle (<https://www.kaggle.com/datasets/hgultekin/bbcnewsarchive>). Выборка содержит 1096 записей при медианном значении длины новостной заметки в символах, равном 2277,5. Тексты подготовлены на английском языке. Особенностью данного набора новостных блоков является типовая структура их построения, в которой сообщается, что комментатор / автор узнал, что что-то произошло или получен какой-то результат значения экономического показателя, счет в матче и т.п.

Для определения будем выбирать одну из категорий (business, sport, entertainment, politics, tech), а задача, как и в другом экспери-

менте с новостями, сводится к задаче бинарной классификации. В результате эксперимента с использованием структур при заданной одинаковой структуре текста для всех категорий их не удается однозначно определить.

Исходя из проведенных экспериментов, можно сформулировать ряд требований к самим текстам. Качество идентификации принадлежности текста к тому или иному классу падает при уменьшении объема этого текста, не зависит от языка и может быть описано характеристикой, приведенной на рис. 46.

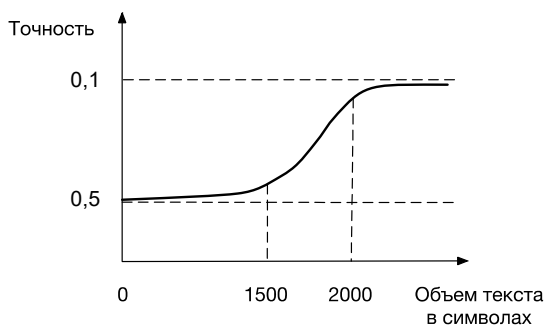


Рис. 46. Изменение точности решения задач классификации в зависимости от объема текста

Эти же эксперименты показывают, что приведенная зависимость может нарушаться, если неправильно выбраны другие характеристики текстовых данных. Такими характеристиками могут быть, например, количество просмотров, после которого текст идентифицируется как популярный, или схожесть текстов. В одном из дополнительных экспериментов задача классификации не решалась, так как новостные заметки, вне зависимости от категории, готовились по единому шаблону.

Таким образом, можно сделать вывод, что предлагаемый нами подход не во всех случаях дает результат, и его следует применять совместно с традиционными подходами, использующими технологии Bag of Words, TF-IDF, Word2Vec, тогда он будет приводить к повышению качества идентификации традиционных подходов,

так как они привязываются к словам и не учитывают их использование после проведения операций токенизации и/или лемматизации, и/или стемминга.

Таким образом, развитием направления реконструкции процессов из текстов является выявление текстов, соответствующих заданному типу, и их структурное описание в форме, позволяющей установить однозначное соответствие между элементами текста и элементами структурно-функционального описания (S, V, NV, R).

ЗАКЛЮЧЕНИЕ

В книге рассмотрен подход, позволяющий исследовать эффективность информационной инфраструктуры организационных систем. Изложенные материалы позволяют обосновывать управленческие решения, связанные с изменениями в информационной инфраструктуре. Получаемые результаты показывают не только предполагаемый итог, но и возможные отклонения. Таким образом может быть решён широкий круг задач, связанный с определением планов и стратегии поведения.

Использование структурного анализа и дизайна ориентировано на решение задач, связанных с изучением существующих процессов, их изменением или заменой на лучшие решения с точки зрения структурной организации или соответствия, используемой в организационной системе методологии управления.

Такое исследование особенно необходимо в условиях увеличивающейся зависимости ПрС от факторов внешней среды, снижения зависимости субъектов бизнеса от базового (основного) потребителя, внедрения новой инновационной продукции.

Введение субъекта управления в явном виде позволяет исследовать, с одной стороны, его влияние на работоспособность всей системы в целом (в том числе с учетом используемой методологии), а с другой стороны, проводить обучение на модели в случаях, когда субъект представлен человеком.

Существующие производственные системы уже используют большое количество специализированных информационных систем в рамках корпоративной информационной системы. Развитие предложенных идей может найти продолжение в анализе взаимодействия этих систем, поиске узких мест и выборе программных решений для модернизации и/или внедрения через анализ процессов / операций / действий, которые они реализуют.

Повышение уровня объективности принимаемых решений становится существенным конкурентным преимуществом при управлении информационной инфраструктурой и организационными системами в целом.

СПИСОК ЛИТЕРАТУРЫ

1. Bennis W.G., Nanus B. Leaders : the strategies for taking charge. NY: Harper&Row Publishers, 1985.
2. Savage S. The flaw of averages // Harv Bus Rev. 2002. Vol. 80, № 11. P. 20–21.
3. Checkland P. Soft systems methodology: a 30-year retrospective. New York: John Wiley, 1999.
4. Cohen M.D., March J.G., Olsen J.P. A Garbage Can Model of Organizational Choice // Adm Sci Q. 1972. Vol. 17, № 1. P. 1–25.
5. Wilden R. et al. 60 Years of March and Simon’s Organizations: An Empirical Examination of its Impact and Influence on Subsequent Research // Journal of Management Studies. 2019. Vol. 56, № 8. P. 1570–1604.
6. Сазерленд Д. SCRUM. Революционный метод управления проектами. М.: Манн, Иванов и Фербер, 2016.
7. Steinke G., Al-Deen M., la Brie R. Innovating Information System Development Methodologies with Design Thinking // Proceedings of 5th International Conference on Applied Innovations in IT / Prof. Dr. Siemens. Koethen: Anhalt University of Applied Sciences, 2017. P. 51–55.
8. Scheer A.-W. ARIS. Modellierungsmethoden, Metamodelle, Anwendungen. 4th ed. Berlin: Springer, 2001. 244 p.
9. Amberg M., Bodendorf F., Möslein K.M. Wertschöpfungsorientierte Wirtschaftsinformatik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. Vol. 4.
10. Dori D. Object-Process Methodology. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.
11. ArchiMate® 3.1 Specification. The Open Group, 2019. 206 p.
12. Питерсон Дж. Теория сетей Петри и моделирования систем. Мир.
13. Forrester J.W. Industrial dynamics // Journal of the Operational Research Society. 1997. № 48(10). P. 1037–1041.
14. van der Aalst W. Process Mining. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

15. Tirole J. The theory of industrial organization. Cambridge, Mass: MIT Press, 1988.
16. Skobelev P. Multi-agent systems for real time resource allocation, scheduling, optimization and controlling: Industrial applications // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2011. Vol. 6867. P. 1–14.
17. Мыльников Л.А. Управление проектами и системами в условиях цифровой экономики. Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2021. 130 с.
18. Мыльников Л.А. Развитие методов и моделей прогнозирования и планирования в задачах управления инновационными проектами в производственно-экономических системах. Пермь: Пермский национальный исследовательский политехнический университет, 2015. 304 с.
19. Voigt K.-I. Industrielles Management: Industriebetriebslehre aus prozessorientierter Sicht. Berlin: Springer, 2008.
20. Кулинич А.А. Компьютерные системы моделирования когнитивных карт: подходы и методы // Проблемы управления. 2010. № 3. С. 2–16.
21. Ширяев В.И., Ширяев Е.В. Управление бизнес-процессами. М.: ИНФРА, 2009.
22. Симонов П.М. Экономико-математическое моделирование. Пермь: Ред.-изд. отд. Пермского гос. ун-та, 2010. 422 с.
23. Немкова Е.А. Математическое моделирование производственных систем с интервальной неопределенностью параметров. Пенза: Пензенский государственный технологический университет, 2014.
24. Симонов П.М. Экономико-математическое моделирование. Пермь: Перм. гос. нац. исслед. ун-т., 2019. Vol. 1. 230 с.
25. Клейнер Г.Б. Производственные функции: теория, методы, применение. М.: Финансы и статистика, 1986. 239 с.
26. Туккель И.Л., Сурина А.В., Культин Н.Б. Управление инновационными проектами. СПб.: БХВ-Петербург, 2011. 416 с.
27. Mylnikov L.A. Structural and Functional Modeling of Processes with a Dedicated Control Subject // Automatic Documentation and Mathematical Linguistics. 2022. Vol. 56, № 1. P. 42–54.

28. Day T. Beyond SMART? A new framework for goal setting // Curric J. 2011. Vol. 22, № 4. P. 515–535.
29. Demartini C. Performance management systems: design, diagnosis and use. Springer.
30. Oslo Manual: Guidelines for Collecting and Interpreting Innovation Data / ed. OECD, Eurostat. OECD / European Communities, 2005.
31. Farrell M.J. The Measurement of Productive Efficiency // Journal of the Royal Statistical Society. 1957. Vol. 120. P. 253–281.
32. Charnes A., Cooper W.W., Rhodes E. Measuring the efficiency of decision-making units // European Journal of Operation Research. 1978. Vol. 2, № 6. P. 429–444.
33. Calmes A. Die Statistik Im Fabrik-Und Warenhandelsbetrieb. FB&C LTD, 2018. 288 p.
34. Gantt H.L., Forrer D. Organizing for work / ed. Forrer D.A. Florida: Donald A. Forrer, 2006.
35. Когаловский В. Происхождение ERP // Директор информационной службы. 2000. № 5.
36. Бурганова Т.А. Управление проектами: гинезис, структура // Международный журнал гуманитарных и естественных наук. 2018. Vol. 12–1. P. 183–187.
37. Мыльников Л.А. Вопросы формирования динамического портфеля инновационных проектов в производственных системах с дискретным временем // XII Всероссийское совещание по проблемам управления. М.: ИПУ РАН, 2019. С. 3134–3138.
38. Inmon W.H., Hackthorn R. Using the Data Warehouse. 1994.
39. Мыльников Л.А. Системный взгляд на проблему моделирования и управления производственными инновациями / Научно-техническая информация. № 5. С. 11–23.
40. Business Process Management Market Size, Share & Trends Analysis Report By Solution (Automation, Process Modelling), By Application, By Deployment, By End-user, By Region, And Segment Forecasts, 2023 – 2030. Grand View Research, 2022. 150 p.

41. Thomas O., Fellmann M.A.M. Semantische Prozessmodellierung – Konzeption und informationstechnische Unterstützung einer ontologiebasierten Repräsentation von Geschäftsprozessen // WIRTSCHAFTSINFORMATIK. 2009. Vol. 51, № 6. P. 506–518.
42. Makhabel B. Learning Data Mining with R. Birmingham: Packt Publishing Ltd, 2015. 314 p.
43. Мыльников Л.А. Статистические методы интеллектуального анализа данных. СПб.: БХВ-Петербург, 2021. 240 p.
44. Bertoni M. et al. PLM paradigm: How to lead BPR within the Product Development field // Comput Ind. 2009. Vol. 60, № 7. P. 476–484.
45. Business Process Model and Notation (BPMN). Object Management Group, 2013. 532 p.
46. El-Saber N., Boronat A. BPMN Formalization and Verification using Maude // Proceedings of the 2014 Workshop on Behaviour Modelling-Foundations and Applications – BM-FA '14. New York, New York, USA: ACM Press, 2014. P. 1–12.
47. Ouyang C. et al. From business process models to process-oriented software systems // ACM Transactions on Software Engineering and Methodology. 2009. Vol. 19, № 1. P. 1–37.
48. Figl K. Comprehension of Procedural Visual Business Process Models // Business & Information Systems Engineering. 2017. Vol. 59, № 1. P. 41–67.
49. Yourdon E., Constantine L.L. Structured design: fundamentals of a discipline of computer program and systems design. Englewood Cliffs, N.J: Prentice Hall, 1979.
50. Evron Y., Soffer P., Zamansky A. Model-based Analysis of Data Inaccuracy Awareness in Business Processes // Business & Information Systems Engineering. 2022. Vol. 64, № 2. P. 183–200.
51. del-Río-Ortega A. et al. Visual ppinot: A Graphical Notation for Process Performance Indicators // Business & Information Systems Engineering. 2019. Vol. 61, № 2. P. 137–161.
52. Modern Business Process Automation / ed. Hofstede A.H.M. et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

53. Koschmider A., Reijers H.A. Improving the process of process modelling by the use of domain process patterns // *Enterp Inf Syst.* 2015. Vol. 9, № 1. P. 29–57.

54. Wang L., Du Y., Qi L. Efficient deviation detection between a process model and event logs // *IEEE/CAA Journal of Automatica Sinica.* 2019. Vol. 6, № 6. P. 1352–1364.

55. Khelif W., Ben-Abdallah H. Integrating semantics and structural information for BPMN model refactoring // *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS).* IEEE, 2015. P. 656–660.

56. Goel K., Bandara W., Gable G. A Typology of Business Process Standardization Strategies // *Business & Information Systems Engineering.* 2021. Vol. 63, № 6. P. 621–635.

57. Первозванский А.А. Математические модели в управлении производством. М.: Наука, 1975. 616 с.

58. Shamir R. The Efficiency of the Simplex Method: A Survey // *Manage Sci.* 1987. Vol. 33, № 3. P. 301–334.

59. Holzschuher F., Peinl R. Performance of Graph Query Languages // *Proceedings of the Joint EDBT/ICDT 2013 Workshops.* Genoa, Italy, 2013. P. 195–204.

60. Awodey S. *Category Theory.* Oxford: Oxford University Press, 2010. 328 p.

61. Петров А.Е.Е. Тензорный метод двойственных сетей. М.: ЦИТИП, 2007. 496 с.

62. Doyle P.G. *Grinstead and Snell's Introduction to Probability.* American Mathematical Society, 2006. 518 p.

63. Mylnikov L. Efficiency management of discrete production systems under the dynamics of project portfolio // *Comput Ind Eng.* 2022. Vol. 163. P. 107807.

64. Amberg M., Mylnikov L. Innovation project lifecycle prolongation method // *Innovation and Knowledge Management in Twin Track Economies Challenges and Solutions – Proceedings of the 11th International Business Information Management Association Conference, IBI-MA 2009.* 2009. Vol. 1–3.

65. Moghaddam K.S. Supplier selection and order allocation in closed-loop supply chain systems using hybrid Monte Carlo simulation and goal programming // *Int J Prod Res.* 2015. № 53(20). P. 6320–6338.

66. Чекал Е.Г., Чичев А.А. Надёжность информационных систем. Ульяновск: УлГУ, 2012. 118 с.

67. Salimifard K., Wright M. Petri net-based modelling of workflow systems: An overview // *Eur J Oper Res.* 2001. Vol. 134, № 3. P. 664–676.

68. Zhongzhi S. Knowledge-Based Decision Support System // *J Comput Sci Technol.* 1987. № 1. Part 2.

69. Сунцов В.П., Мыльников Л.А. Исследование информационных процессов управления разработкой строительной документации // *Информационные ресурсы России.* 2022. № 5–6. С. 40–47.

70. Мыльников Л.А. Структурно-функциональное моделирование в управлении эффективностью информационной инфраструктуры поддержки принятия решений производственных систем // *Научно-техническая информация. Серия 2: Информационные процессы и системы.* 2023. № 1. С. 10–22.

71. Fahrmeir L., Kuenstler R., Pigeot I. Statistik (Der Weg zur Datenanalyse). Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. 610 p.

72. Hull T.E., Dobell A.R. Random Number Generators // *SIAM Review.* 1962. Vol. 4, № 3. P. 230–254.

73. Carmichael R.D. On the Numerical Factors of the Arithmetic Forms $\alpha n \pm \beta n$ // *The Annals of Mathematics.* 1913. Vol. 15, № 1/4. P. 30.

74. Box G.E.P., Muller M.E. A Note on the Generation of Random Normal Deviates // *The Annals of Mathematical Statistics.* 1958. Vol. 29, № 2. P. 610–611.

75. Teichroew D. The Mixture of Normal Distributions with Different Variances // *The Annals of Mathematical Statistics.* 1957. Vol. 28, № 2. P. 510–512.

76. Мыльников Л.А., Трушников Д.Н. Моделирование систем: практикум. Пермь: ПГТУ, 2006.

77. Novikova A. Direct Machine Translation and Formalization Issues of Language Structures and Their Matches by Automated Machine

Translation for the Russian-English Language Pair // Proceedings of International Conference on Applied Innovation in IT. 2018. Vol. 6, № 1. P. 85–92.

78. Новикова А.В. Лингвистический анализ реализации возможных миров в художественном тексте. Челябинск: Челябинский государственный университет, 2010.

79. Austin J.L., Urmson J.O. How to Do Things with Words. Cambridge, Massachusetts: Harvard University Press, 2009.

80. Searle J.R. Speech Acts, An Essay in the Philosophy of Language. Cambridge, Massachusetts: Cambridge University Press, 1969.

81. Habermas J. Entgegnung // Kommunikatives Handeln. Beiträge zu Jürgen Habermas' «Theorie des kommunikativen Handelns». Frankfurt am Main: Suhrkamp Verlag, 1986. P. 327–406.

82. Habermas J. Theorie des kommunikativen // Handlungsrationalität und gesellschaftliche Rationalisierung. Frankfurt am Main: Suhrkamp Verlag, 1981.

83. Арутюнова Н.Д. Истина: фон и коннотации // Логический анализ языка. Культурные концепты / под ред. Н.Д. Арутюнова, Т.Е. Янко М.: Наука, 1991. С. 21–30.

84. Блок А.А. Полное собрание сочинений в 20 т., Т.5. М.: Наука, 1999.

85. Чаплыгина Т.Л. Лирика Арсения Тарковского в контексте поэзии Серебряного века. Ивановский государственный университет, 2007.

86. Дзущева Н.В. Русская поэзия 1910–1920-х годов в аспекте постсимволизма (Проблемы эстетики и поэтики). Иваново: Ивановский государственный университет, 1999.

87. Арутюнова Н.Д. Проблемы синтаксиса и семантики в работах Ч. Филмора // Вопросы языкознания. 1973. № 1. С. 123.

88. Арутюнова Н.Д. Понятие пресуппозиции в лингвистике // Известия АН СССР: Серия литературы и языка. Т. XXXII, № 1. С. 89.

89. Mayakovsy Vladimir 1894–1930. The bedbug [a play] and selected poetry / ed. Blake P. Indiana University Press & Bloomington and Indianapolis, 1960.

90. Петрушевская Л. Пуськи бятые. Бурлак. Пуськинисты // Искусство кино. 1997. № 3.

91. Rong X. word2vec Parameter Learning Explained [Электронный ресурс]. 2016. Р. <https://arxiv.org/pdf/1411.2738.pdf>.

92. Трусов В.А. Концептуальный подход к поиску и семантической обработке научно-технической информации в распределенных системах интернета // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2021. № 4. С. 1–11.

93. Мыльникова А.В., Мыльников Л.А. Вопросы дистрибутивно-смыслового анализа скелетных структур текстов в задачах автоматизированной обработки языковых данных // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2023. № 5. С. 21–30.

Научное издание

МЫЛЬНИКОВ Леонид Александрович,

АВРАМОВИЧ Зоран,

МЫЛЬНИКОВА Анна Вячеславовна

ИССЛЕДОВАНИЕ
ИНФОРМАЦИОННЫХ ПРОЦЕССОВ
ОРГАНИЗАЦИОННЫХ СИСТЕМ

Монография

Подписано в печать 11.12.2023. Формат 60×90/16.

Усл. печ. л. 11,3. Тираж 28 экз. Заказ № 287.

Издательство

Пермского национального исследовательского
политехнического университета.

Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113.

Тел. (342) 219-80-33.

Отпечатано в типографии Издательства

Пермского национального исследовательского
политехнического университета.

Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113.

Тел. (342) 219-80-33.