

Ю. Ю. Громов, В. Е. Дидрих, О. Г. Иванова, В. Г. Однолько
ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Ю. Ю. ГРОМОВ, В. Е. ДИДРИХ,
О. Г. ИВАНОВА, В. Г. ОДНОЛЬКО

ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Тамбов
◆ Издательство ФГБОУ ВПО «ТГТУ» ◆
2014

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тамбовский государственный технический университет»

**Ю. Ю. ГРОМОВ, В. Е. ДИДРИХ,
О. Г. ИВАНОВА, В. Г. ОДНОЛЬКО**

ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебника для студентов высших учебных заведений,
обучающихся по направлению 230400
«Информационные системы и технологии»



Тамбов
Издательство ФГБОУ ВПО «ТГТУ»
2014

УДК 004.89(075)
ББК 32.813я73
Т338

Рецензенты:

Доктор физико-математических наук, профессор института
проблем экоинформатики Академии естественных наук РФ
В. Ф. Крапивин

Кандидат технических наук, профессор ФГБОУ ВПО «ТГТУ»
Ю. Ф. Мартемьянов

Т338 **Теория** информационных процессов и систем : учебник /
Ю. Ю. Громов, В. Е. Дидрих, О. Г. Иванова, В. Г. Однолько. –
Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2014. – 172 с. –
100 экз. – ISBN 978-5-8265-1352-1

Рассмотрены основные модели и методы теории систем и систем-
ного анализа.

Предназначен для студентов высших учебных заведений, обу-
чающихся по направлению подготовки 230400 «Информационные
системы и технологии».

УДК 004.89(075)
ББК 32.813я73

ISBN 978-5-8265-1352-1

© Федеральное государственное бюджетное
образовательное учреждение высшего
профессионального образования
«Тамбовский государственный технический
университет» (ФГБОУ ВПО «ТГТУ»), 2014

ВВЕДЕНИЕ

Понятие «система» в буквальном переводе с греческого означает «целое, составленное из частей». В настоящее время термин «система» часто употребляется в различных сферах деятельности человека и в разных предметных областях, прежде всего, благодаря очень эффективному стилю мышления в рамках сложившейся методологии системного подхода. Системное мышление – это мышление современного человека. Системный стиль мышления, или системный подход представляет собой специфическое содержание, аспект, принцип мышления, при котором категория «система» применяется в качестве метода, инструмента познания.

Термин «системный подход» определяет группу методов, с помощью которых реальный объект моделируется совокупностью взаимодействующих элементов. Эти методы развиваются в рамках отдельных научных дисциплин и общенаучных концепций, являются результатом их междисциплинарного синтеза. Успешное применение системного подхода в науке определяется успехом частных системных теорий в других областях знаний, в частности эффективным развитием кибернетики.

Системный подход как эффективный способ мыслительной деятельности обеспечил значительные открытия в науке, изобретения в технике и достижения в производстве во второй половине XX века. Это предопределяет постоянное внимание к нему со стороны интеллектуалов. Очевидно, что потребность в системном мышлении диктует необходимость изучения в высших учебных заведениях дисциплин, основанных на теории систем и связанных с системным анализом, поскольку без владения этим методами системного подхода невозможны эффективная профессиональная деятельность и тем более творческая самореализация.

Объектом осмысления в учебнике являются системы, а предметом – основные модели и методы теории систем и системного анализа.

Особенность применения методов системного анализа к объекту «Информационная система» (ИС) заключается в том, что информационная система при этом может выступать как реально существующий объект или как проектируемый объект.

Владение системным анализом, системным моделированием и проектированием, системной практической деятельностью – высшая характеристика мыслительной культуры человека, имеющего высоко-профессиональный уровень подготовки. Кроме того, сегодняшнему

специалисту часто приходится заниматься систематизацией информации, системными исследованиями, которые можно осуществлять, только обладая специальными знаниями и навыками. Особенно это актуально для специалистов в области информационных систем и технологий, информационной безопасности.

В настоящее время развитие системного анализа, прежде всего, характеризуется осмыслением широчайшего проникновения вычислительной техники в процесс принятия решения в сложной системе. Программные и технические средства различного уровня и масштаба выполняют значительное число отдельных процедур и начинают эффективно использоваться для составления наборов процедур и контроля за ходом решения задачи в целом. Особое место при анализе и принятии решения занимают такие сравнительно новые объекты, как информационная база (банки данных), диалоговые системы, имитационное моделирование. Эти объекты, обычно воспринимаемые как части автоматизированных систем или как специальные, использующие ЭВМ методы исследования, могут и должны рассматриваться и в качестве важных понятий системного анализа. Они отражают существенные и достаточно абстрактные стороны современного состояния аппаратной реализации системных исследований. С точки зрения системного анализа это некоторые классы операций, обладающие внутренней структурой, универсальностью использования и другими особенностями.

Разнообразная считающая, управляющая, хранящая, преобразующая, советующая, изображающая и другая вычислительная техника является неотъемлемой частью как самих сложных систем, так и их исследования. Владение аппаратом системного анализа невозможно без умения определять тактику и стратегию использования ЭВМ, баз данных вычислительных сетей. В конкретных же проблемах это умение часто вообще определяет успех системного исследования.

Как известно, информационная система (ИС) это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности ИС. Современные крупные проекты ИС характеризуются, как правило, следующими особенностями:

- сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между

ними), требующая тщательного моделирования и анализа данных и процессов;

- наличие совокупности тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования (например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, и приложений аналитической обработки (поддержки принятия решений), использующих нерегламентированные запросы к данным большого объёма);

- ограниченная возможность использования каких-либо типовых проектных решений и прикладных систем;

- необходимость интеграции существующих и вновь разрабатываемых приложений;

- функционирование в неоднородной среде на нескольких аппаратных платформах;

- разобщённость и разнородность отдельных групп разработчиков по уровню квалификации и сложившимся традициям использования тех или иных инструментальных средств;

- существенная временная протяжённость проекта, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков, и, с другой стороны, масштабами организации-заказчика и различной степенью готовности отдельных её подразделений к внедрению ИС.

Эти особенности красноречиво свидетельствуют о справедливости применения аппарата теории систем к ИС.

Процессы, обеспечивающие работу информационной системы любого назначения:

- ввод информации из внешних или внутренних источников;

- обработка входной информации и представление её в удобном виде;

- вывод информации для представления потребителям или передачи в другую систему;

- обратная связь – это информация, переработанная людьми данной организации для коррекции входной информации.

Бизнес-процесс организации – более общее понятие, может включать в себя процессы информационной системы, но выходить за её пределы.

Информационный поток – это совокупность циркулирующих в системе, между системой и внешней средой сообщений, необходимых для управления, анализа и контроля операций. Информационный поток может объединять носители разных видов, например, состоять из бумажных и электронных носителей, которые дублируют или допол-

няют друг друга. Структура информационных потоков определяет их однородность и неоднородность. Однородные информационные потоки характеризуются единым видом носителя, единой функциональной принадлежностью. По периодичности информационные потоки делятся на регулярные, соответствующие регламентированной во времени передаче данных, и оперативные – обеспечивающие связь по требованию.

Информационный поток характеризуется следующими показателями:

- источник возникновения;
- направление движения потока;
- скорость передачи и приёма;
- интенсивность потока и др.

Формирование информационных систем невозможно без исследования потоков в разрезе определённых показателей. Если скорость и интенсивность потока не важны, критичным может оказаться их тип (видео, звук, печатные документы определённых видов).

Поэтому цель учебника сводится к тому, чтобы методически лаконично представить знание о системах в аспекте информационных систем и процессов, выделить и осмыслить тенденции её развития в условиях интенсивного применения информационных технологий.

В данном учебнике использованы положения и материалы, в том числе методические подходы многих ведущих учёных рассматриваемой предметной области, например: работы В. Г. Афанасьева, В. М. Глушкова, Ю. Ю. Громова, В. П. Кузьмина, Ю. Г. Маркова, И. Б. Новика, Л. А. Петрушенко, В. Н. Садовского, М. И. Сетрова, В. Н. Спицнаделя, В. С. Тюхтина, А. И. Умова, Э. Г. Юдина, В. Н. Волковой, А. А. Денисова, Б. Я. Советова и др., предлагаемые ими в научных трудах, учебниках и учебных пособиях по теории систем и смежным научным направлениям.

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

1.1. ИСТОРИЧЕСКАЯ СПРАВКА

Первый вариант общей теории систем был предложен в 1912 г. А. А. Богдановым (настоящая фамилия Малиновский; 1873 – 1928) в виде учения, названного тектологией. Тектология Богданова – это общая теория организации и дезорганизации, наука об универсальных типах и закономерностях структурного преобразования любых систем. Несомненно, что А. А. Богданову удалось заложить основы новой синтетической науки, хотя и не получавшей признания длительное время.

Основная идея тектологии состоит в тождественности организации систем разных уровней: от микромира до биологических и социальных систем. Относительно социальных процессов А. А. Богданов считал, что всякая человеческая деятельность объективно является организующей или дезорганизующей. Он полагал, что дезорганизация – частный случай организации. Во всём мире происходит борьба организационных форм, и в ней побеждают более организованные формы (неважно, идёт ли речь об экономике, политике, культуре или идеологии). Это происходит из-за того, что организационная система всегда больше, чем сумма её составляющих элементов, а дезорганизационная – всегда меньше суммы своих частей. Поэтому главная задача тектологии заключается в лучшей организации вещей (техники), людей (экономики) и идей.

А. А. Богданов одним из первых в мире ввёл понятие системности. Состояние системы определяется равновесием противоположностей. В результате непрерывного взаимодействия формируются три вида систем, которые он подразделяет на организованные, неорганизованные и нейтральные.

Учёный разработал идею о структурной устойчивости системы и её условиях. В самой системе одним из первых увидел два вида закономерностей:

а) *формирующие*, т.е. закономерности развития, приводящие к переходу системы в другое качество;

б) *регулирующие*, т.е. закономерности функционирования, способствующие стабилизации нынешнего качества системы.

Он ввёл также ряд интересных понятий, характеризующих этапы развития различных систем, к ним относятся следующие:

– «комплексия» употреблялся им для обозначения ситуации, когда система представляет собой чисто механическое объединение элементов, между которыми ещё не начались процессы взаимодействия. Это характерно для случаев, когда, скажем, предприниматель начинает создавать организацию (набрал кадры, закупил технику, нанял помещение и т.д.), но она ещё не функционирует;

– «конъюгация» означает такой этап развития системы, когда начинается сотрудничество между её отдельными элементами системы (например, работники установили между собой формальные и неформальные отношения);

– «ингрессия» выражает этап перехода системы к новому качеству (например, рост сплочённости, взаимопонимания, сработанности коллектива), а понятие «дезингрессия», наоборот, означает процесс деградации системы, её распада как целостного объединения.

В Берлине А. А. Богданов опубликовал свои идеи. С ними ознакомился австрийский биолог и философ Людвиг фон Берталанфи (1901 – 1972). Он и создал второй вариант общей теории систем. В 30 – 40-е годы Берталанфи заложил основы концепции организмического подхода к организованным динамическим системам, обладающим свойством эквифинальности, т.е. способностью достигать цели независимо от нарушений на начальных этапах развития. Он обобщил принципы целостности, организации и изоморфизма в единую концепцию. Сначала применил идею открытых систем к объяснению ряда проблем биологии и генетики, но потом пришёл к выводу, что методология системного подхода является более широкой и может быть применима в различных областях науки. Так возникла идея общей теории систем. Идеи Берталанфи привлекли внимание международной научной общественности, а идеи Богданова оказались невостребованным потенциалом науки. Это тот, почти библейский случай, когда идеи, как зёрна: одни упали на неподготовленную почву, а другие – на благодатную.

Л. Берталанфи сыграл огромную роль в становлении и популяризации системного подхода. Он первым поставил саму задачу построения этой теории. Общая теория систем мыслилась им как фундаментальная наука, исследующая проблемы систем различной природы.

Существенный недостаток в понимании Л. Берталанфи общей теории систем состоял в том, что он объявил её заменяющей философию, что вызвало справедливые возражения философов. Если обратить внимание на содержание общей теории систем, то в неё входят

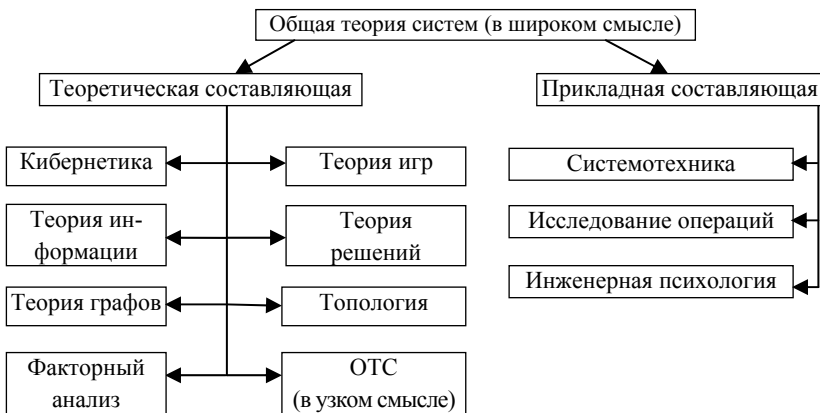


Рис. 1.1. Схема общей теории систем в представлении Л. Берталанфи

в основном формализованные науки, которые хорошо применимы к относительно простым системам. Потребность исследования сложных систем заставляет использовать качественный анализ, которым владеют философские науки. Но философии систем в общей теории систем места не нашлось. Поэтому произошло раздвоение общей теории систем (ОТС) на ОТС в широком смысле и на ОТС в узком (рис. 1.1).

Сама ОТС в узком смысле также в значительной мере осталась количественно-формальной наукой.

Последующее развитие системных знаний привело к тому, что возникло несколько вариантов общей теории систем в узком смысле слова, сформировалось знание, которое отражало отдельные стороны систем, появились значительные наработки о системах различной природы: физических, химических, биологических, психических и социальных.

В качестве особого и главного популяризатора системных идей выступила научно-техническая революция, которая обеспечила бурное развитие системного подхода. На Западе идеи теории систем развивали такие учёные, как Р. Акофф, О. Ланге, Р. Мертон, М. Месарович, Т. Парсонс, У. Росс Эшби и др. В СССР в 60 – 70-е годы проблемы системологии, создания общей теории систем были также очень популярными. Исследованиями здесь занимались В. Г. Афанасьев, В. М. Глушков, В. П. Кузьмин, Ю. Г. Марков, И. Б. Новик, Л. А. Петрушенко, В. Н. Садовский, М. И. Сетров, В. С. Тюхтин, А. И. Уемов, Э. Г. Юдин и другие учёные.

Из схемы ОТС (рис. 1.1) очевидно, что эта методология как теоретическими составляющими, так и практическими связана с современными информационными системами и процессами не только в аспекте их анализа и синтеза, а также в аспекте применения информационных технологий с целью реализации моделей и методов теории систем.

1.2. ТЕРМИНОЛОГИЯ ТЕОРИИ СИСТЕМ

Термин «система» употребляется во множестве различных смысловых вариаций.

Система – это теория (например, философская система Платона). По всей видимости, этот контекст понимания системы был наиболее ранним – как только возникли первые теоретические комплексы. И чем универсальнее они были, тем больше была потребность в специальном термине, который обозначал бы эту целостность и универсальность.

Система – это классификация (например, периодическая система элементов Д. И. Менделеева). Особенно бурно возникали различные классификационные системы в XVIII – XIX веках. Основная проблема классификаций заключается в том, чтобы они были существенными и не систематизировали объекты с точки зрения несущественных признаков.

Система – это завершённый метод практической деятельности (например, система реформатора театра К. С. Станиславского). Такого рода системы складывались по мере возникновения профессий, накопления профессиональных знаний и навыков.

Система – некоторый способ мыслительной деятельности (например, система исчисления). Этот вид системы имеет древние истоки. Они начинались с систем письма и исчисления и развились до информационных систем современности.

Система – это совокупность объектов природы (например, Солнечная система). Натуралистическое употребление термина связано с автономностью, некоторой завершённостью объектов природы, их единством и целостностью.

Система – это некоторое явление общества (например, экономическая система, правовая система). Социальное употребление термина обусловлено непохожестью и разнообразием человеческих обществ, формированием их составляющих: правовой, управленческой, социальной и других систем.

Система – это совокупность установившихся норм жизни, правил поведения. Речь идёт о некоторых нормативных системах, которые свойственны различным сферам жизни людей и общества (например,

законодательная и моральная), выполняющих регулятивную функцию в обществе.

Таким образом, анализ многообразия употребления понятия «система» показывает, что оно имеет древние корни и играет очень важную роль в современной культуре, выступает интегралом современного знания, средством постижения всего сущего. Вместе с тем понятие не однозначно и не жёстко, что делает его исключительно креативным.

Очевидно, что в настоящее время нет единства в определении понятия «система». В первых определениях в той или иной форме говорилось о том, что система – это элементы и связи (отношения) между ними. Например, основоположник теории систем Людвиг фон Бергланфи определял систему как комплекс взаимодействующих элементов или как совокупность элементов, находящихся в определённых отношениях друг с другом и со средой. А. Холл определяет *систему* как множество предметов вместе со связями между предметами и между их признаками. Ведутся дискуссии, какой термин – «отношение» или «связь» – лучше употреблять.

Позднее в определениях системы появляется понятие цели. Так, в Философском словаре «система» определяется как «совокупность элементов, находящихся в отношениях и связях между собой определённым образом и образующих некоторое целостное единство».

В последнее время в определение понятия системы наряду с элементами, связями и их свойствами и целями начинают включать наблюдателя, хотя впервые на необходимость учёта взаимодействия между исследователем и изучаемой системой указал один из основоположников кибернетики У. Р. Эшби.

М. Масарович и Я. Такахага в книге «Общая теория систем» считают, что система – «формальная взаимосвязь между наблюдаемыми признаками и свойствами».

Таким образом, в зависимости от количества учитываемых факторов и степени абстрактности определение понятия «система» можно представить в следующей символической форме (концептуальной модели). Каждую концепцию обозначим буквой D (от лат. *definitions*) и порядковым номером, совпадающим с количеством учитываемых в определении факторов.

D_1 . Система есть нечто целое. $S = A(1, 0)$. Это определение выражает факт существования и целостность.

Двоичное суждение $A(1, 0)$ отображает наличие или отсутствие этих качеств.

D_2 . Система есть организованное множество: $S = (Org, M)$, где Org – оператор организации; M – множество (подразумеваются составные части).

D_3 . Система есть множество вещей, свойств и отношений: $S = (\{m\}, \{n\}, \{r\})$, где m – вещи, n – свойства, r – отношения.

D_4 . Система есть множество элементов, образующих структуру и обеспечивающих определённое поведение в условиях окружающей среды: $S = (\varepsilon, ST, BE, E)$, где ε – элементы, ST – структура, BE – поведение, E – среда.

D_5 . Система есть множество входов, множество выходов, множество состояний, характеризуемых оператором переходов и оператором выходов: $S = (X, Y, Z, H, G)$, где X – входы, Y – выходы, Z – состояния, H – оператор переходов, G – оператор выходов. Это определение учитывает все основные компоненты, рассматриваемые в автоматике.

D_6 . Это шестичленное определение соответствует уровню биосистем и учитывает генетическое (родовое) начало GN , условия существования KD , обменные явления MB , развитие EV , функционирование FC и репродукцию (воспроизведения) RP : $S = (GN, KD, MB, EV, FC, RP)$.

D_7 . Это определение оперирует понятиями модели F , связи SC , пересчёта R , самообучения FL , самоорганизации FQ , проводимости связей CO и возбуждения моделей JN : $S = (F, SC, R, FL, FQ, CO, JN)$. Данное определение удобно при нейрокибернетических исследованиях.

D_8 . Если определение D_5 дополнить фактором времени и функциональными связями, то получим определение системы, которым обычно оперируют в теории автоматического управления: $S = (T, X, Y, Z, V, V_z, F, f)$, где T – время, X – входы, Y – выходы, Z – состояния, V – класс операторов на выходе, V_z – значения операторов на выходе, F – функциональная связь в уравнении $y(t_2) = F(x(t_1), z(t_1), t_2)$, f – функциональная связь в уравнении $z(t_2) = f(x(t_1), z(t_1), t_2)$.

D_9 . Для организационных систем удобно в определении системы учитывать следующее:

$S = (PL, RO, RJ, EX, PR, DT, SV, RD, EF)$, где PL – цели и планы, RO – внешние ресурсы, RJ – внутренние ресурсы, EX – исполнители, PR – процесс, DT – помехи, SV – контроль, RD – управление, EF – эффект.

Последовательность определений можно продолжить до D_n ($n = 10, 11, \dots$), при этом учитывалось бы такое количество элементов, связей и действий в реальной системе, которое необходимо для решаемой задачи, для достижения поставленной цели.

В качестве рабочего определения понятия системы в литературе по теории систем часто рассматривается следующее. Система – мно-

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

Из схемы ОТС (рис. 1.1) очевидно, что эта методология тесно связана с современными информационными системами, причём как в аспекте их анализа и синтеза, так и в аспекте применения информационных технологий к реализации моделей и методов системного подхода.

На практике, особенно начинающими системными аналитиками, наиболее часто применяется концептуальная модель D_3 : $S = (\{m\}, \{n\}, \{r\})$, в которой в отличие от трактовки Уйменова А. И. параметры модели задаются несколько другими терминами без изменения их сущности, так m – элементы системы, n – свойства, r – связи (отношения). При этом свойства могут рассматриваться как в отношении элементов n_m , так и в отношении связей n_r , а связи в свою очередь могут быть как между элементами r_m , так и между свойствами элементов r_n .

С одной стороны, интегративное свойство системы обеспечивает её целостность, качественно новое образование по сравнению с составляющими её частями, а с другой – под системой понимается объект, свойства которого не сводятся без остатка к свойствам составляющих его дискретных элементов (*неаддитивность свойств*).

Под элементом принято понимать простейшую неделимую часть системы. Ответ на вопрос, что является такой частью, может быть неоднозначным и зависит от цели рассмотрения объекта как системы, от точки зрения на него или от аспекта его изучения. Таким образом, элемент – это предел деления системы с точек зрения решения конкретной задачи и поставленной цели. Систему можно расчленить на элементы различными способами в зависимости от формулировки цели и её уточнения в процессе исследования. Любой элемент системы можно рассматривать как самостоятельную систему (математическую модель, описывающую какой-либо функциональный блок, или аспект изучаемой проблемы), как правило, более низкого порядка. Каждый элемент системы описывается своей функцией. Под *функцией* понимаются присущие живой и костной материи вещественно-энергетические и информационные отношения между входными и выходными процессами. Если такой элемент обладает внутренней структурой, то его называют *подсистемой*, такое описание может быть использовано при реализации методов анализа и синтеза систем. Это нашло отражение в одном из принципов системного анализа – *законе системности*, говорящем о том, что любой элемент может быть либо подсистемой в некоторой системе, либо подсистемой среди множества объектов аналогичной категории. Элемент всегда является частью системы и вне её не представляет смысла.

Система может быть разделена на элементы не сразу, а последовательным расчленением на подсистемы, которые представляют собой компоненты более крупные, чем элементы, и в то же время более детальные, чем система в целом. Возможность деления системы на подсистемы связана с вычленением совокупностей взаимосвязанных элементов, способных выполнять относительно независимые функции, подцели, направленные на достижение общей цели системы. Названием «подсистема» подчёркивается, что такая часть должна обладать свойствами системы (в частности, свойством целостности). Этим подсистема отличается от простой группы элементов, для которой не сформулирована подцель и не выполняются свойства целостности (для такой группы используется название «компоненты»).

Понятие «структура» происходит от латинского слова *structure*, означающего строение, расположение, порядок. Структура отражает наиболее существенные взаимоотношения между элементами и их группами (компонентами, подсистемами), которые мало меняются при изменениях в системе и обеспечивают существование системы и её основных свойств. *Структура* – это совокупность элементов и связей между ними. Структура может быть представлена графически, в виде теоретико-множественных описаний, матриц, графов и других языков моделирования структур.

Структуру часто представляют в виде иерархии. *Иерархия* – это упорядоченность компонентов по степени важности (многоступенчатость, служебная лестница). Между уровнями иерархической структуры могут существовать взаимоотношения строгого подчинения компонентов (узлов) нижележащего уровня одному из компонентов вышележащего уровня, т.е. отношения так называемого древовидного порядка. Такие иерархии называют сильными или иерархиями типа «дерева». Они имеют ряд особенностей, делающих их удобным средством представления систем управления. Однако могут быть связи и в пределах одного уровня иерархии. Один и тот же узел нижележащего уровня может быть одновременно подчинён нескольким узлам вышележащего уровня. Такие структуры называют иерархическими структурами «со слабыми связями». Между уровнями иерархической структуры могут существовать и более сложные взаимоотношения, например, типа «страт», «слоёв», «эшелонов».

Понятие «связь» входит в любое определение системы наряду с понятием «элемент» и обеспечивает возникновение и сохранение структуры и целостных свойств системы. Это понятие характеризует одновременно и строение (статику), и функционирование (динамику) системы.

Связь характеризуется направлением, силой и характером (или видом). По первым двум признакам связи можно разделить на направленные и ненаправленные, сильные и слабые, а по характеру – на связи подчинения, генетические, равноправные (или безразличные), связи управления. Связи можно разделить также по месту приложения (внутренние и внешние), по направленности процессов в системе в целом или в отдельных её подсистемах (прямые и обратные). Связи в конкретных системах могут быть одновременно охарактеризованы несколькими из названных признаков.

Важную роль в системах играет понятие «обратной связи». Обратная связь является основой саморегулирования и развития систем, приспособления их к изменяющимся условиям существования.

Понятием «состояние» обычно характеризуют мгновенную фотографию, «срез» системы, остановку в её развитии. Его определяют либо через входные воздействия и выходные сигналы (результаты), либо через макропараметры, макросвойства системы (например, давление, скорость, ускорение – для физических систем; производительность, себестоимость продукции, прибыль – для экономических систем).

Более полно состояние можно определить следующей формальной моделью: если рассмотреть элементы ε (или компоненты, функциональные блоки), определяющие состояние, учесть, что «входы» можно разделить на управляющие u и возмущающие x (неконтролируемые), а «выходы» (выходные результаты, сигналы) зависят от ε , u и x , т.е. $z_t = f(\varepsilon_t, u_t, x_t)$, тогда в зависимости от задачи состояние может быть определено как $\{\varepsilon, u\}$, $\{\varepsilon, u, z\}$ или $\{\varepsilon, x, u, z\}$. Таким образом, *состояние* – это множество существенных свойств, которыми система обладает в данный момент времени.

Если система способна переходить из одного состояния в другое (например, $z_1 \rightarrow z_2 \rightarrow z_3$), то говорят, что она обладает «поведением». Этим понятием пользуются, когда неизвестны закономерности переходов из одного состояния в другое. Тогда говорят, что система обладает каким-то поведением, и выясняют его закономерности. С учётом введённых выше обозначений поведение можно представить как функцию $z_t = f(z_{t-1}, u_t, x_t)$.

Под *внешней средой* понимается множество элементов, которые не входят в систему, но изменение их состояния вызывает изменение поведения системы.

Под *моделью* системы понимается описание системы, отображающее определённую группу её свойств. Углубление описания – *детализация модели*. Создание модели системы позволяет предсказывать её поведение в определённом диапазоне условий.

Модель функционирования (поведения) системы – это модель, предсказывающая изменение состояния системы во времени, например: натурные (аналоговые), математические, имитационные, машинные на ЭВМ и др.

Типы поведения системы определяются способностями системы и процессами в системе.

Равновесие системы – это способность системы в отсутствие внешних возмущающих воздействий (или при постоянных воздействиях) сохранить своё состояние сколь угодно долго.

Устойчивость системы – это способность системы возвращаться в состояние равновесия после того, как она была из этого состояния выведена под влиянием внешних возмущающих воздействий. Эта способность обычно присуща системам при постоянном воздействии и если только отклонения не превышают некоторого предела.

Состояние равновесия, в которое система способна возвращаться, по аналогии с техническими устройствами называют *устойчивым состоянием равновесия*. Равновесие и устойчивость в экономических и организационных системах – гораздо более сложные понятия, чем в технике, и до недавнего времени ими пользовались только для некоторого предварительного описательного представления о системе. В последнее время появились попытки формализованного отображения этих процессов и в сложных организационных системах, помогающие выявлять параметры, влияющие на их протекание и взаимосвязь.

Исследованию процесса *«развития»*, соотношения процессов развития и устойчивости, изучению механизмов, лежащих в их основе, уделяют в кибернетике и теории систем большое внимание. Понятие развития помогает объяснить сложные термодинамические и информационные процессы в природе и обществе.

Применение понятия *«цель»* и связанных с ним понятий целенаправленности, целеустремленности, целесообразности сдерживается трудностью их однозначного толкования в конкретных условиях. Это связано с тем, что процесс целеобразования и соответствующий ему процесс обоснования целей в организационных системах весьма сложен и не до конца изучен. Его исследованию большое внимание уделяется в психологии, философии, кибернетике. В Большой Советской Энциклопедии цель определяется как «заранее мыслимый результат сознательной деятельности человека». В практических применениях *цель* – это идеальное устремление, которое позволяет коллективу увидеть перспективы или реальные возможности, обеспечивающие своевременность завершения очередного этапа на пути к идеальным устремлениям.

В настоящее время в связи с усилением программно-целевых принципов в планировании исследованию закономерностей целеобразования и представления целей в конкретных условиях уделяется всё больше внимания. Понятие «цель» лежит в основе развития системы.

Существует два принципиально разных подхода к определению системы: дескриптивный и конструктивный. Рассмотрим их специфику.

Дескриптивный подход основан на признании того, что системность свойственна действительности, что окружающий мир, Вселенная представляют собой некоторую совокупность систем, всеобщую систему систем, что каждая система принципиально познаваема, что внутри системы существует неслучайная связь между её элементами, структурой и функциями, которые эта система выполняет.

Отсюда дескриптивный подход к системе заключается в том, что характер функционирования системы объясняют её структурой, элементами, что находит отражение в определениях системы, которые называются дескриптивными. К ним относятся почти все определения, которые анализировались ранее. В соответствии с дескриптивным подходом, любой объект выступает как система, но только в том аспекте, в каком его внешнее проявление (свойство, функция) задаётся его внутренним устройством (отношением, структурой, взаимосвязями). Идеология этого подхода проста: всё в мире есть системы, но лишь в определённом отношении.

Дескриптивный подход лежит в основе системного анализа, который состоит в том, что обоснованно выделяется и осмысливается структура системы, из которой выводятся её функции. Схема может быть такой:

- выделение элементов, имеющих некоторую пространственно-временную определенность;
- определение связей между элементами;
- определение системообразующих свойств, связей и отношений; определение структур, т.е. законов композиции; анализ функций системы.

Конструктивный подход носит обратный характер. В нём по заданной функции конструируется соответствующая ей структура. При этом используется не просто функциональный, но и функционально-целевой подход, потому что система должна соответствовать некоторым целям конструирования. Выделение и построение системы осуществляются так:

- ставится цель, которую должна выполнить система; определяется функция (или функции), обеспечивающая(ие) достижение этой цели;
- подыскивается или создаётся структура, обеспечивающая выполнение функции.

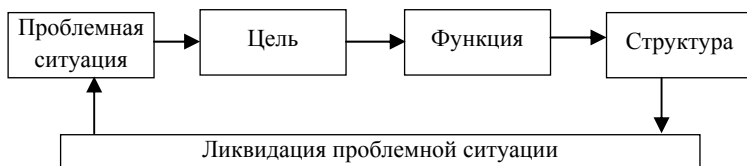


Рис. 1.2. Конструирование системы

Цель представляет собой состояние, к которому направлена тенденция движения объекта. В неживой природе существуют объективные цели, а в живой дополнительно – субъективные цели. Образно говоря, объективная цель – это мишень для поражения, а субъективная цель – желание стрелка поразить её. Цель обычно возникает из проблемной ситуации, которая не может быть разрешена наличными средствами. И система выступает средством разрешения проблемы. Схематично это представлено на рис. 1.2.

Теперь дадим следующее конструктивное определение системы. Система есть конечное множество функциональных элементов и отношений между ними, выделяемое из среды в соответствии с заданной целью в рамках определённого временного интервала.

1.3. ПОДХОДЫ К КЛАССИФИКАЦИИ СИСТЕМ

Классификация систем представляет собой исключительно сложную проблему, которая ещё не разрешена в науке. Одна из причин заключается в том, что конкретных разновидностей систем столь много, что создаётся ощущение их полного совпадения со всеми типами имеющихся объектов. Другая причина состоит в абстрактности понимания самой системы. К тому же до сих пор не выработаны общие параметры (классификационные признаки), характеризующие систему.

Одной из первых попыток создания классификации систем была попытка А. А. Богданова. В результате непрерывного взаимодействия формируется три вида комплекса (системы), которые Богданов различает по степени их организованности – организованные, неорганизованные, нейтральные.

Ныне существуют самые разнообразные подходы к классификации систем. Б. А. Гладких с соавторами анализирует классификации видов, представленных на рис. 1.3.

Предметная классификация строится на основе выделения всех видов конкретных систем. Такова, например, классификация Стефана Бира, которая представляет собой матрицу (табл. 1.1).

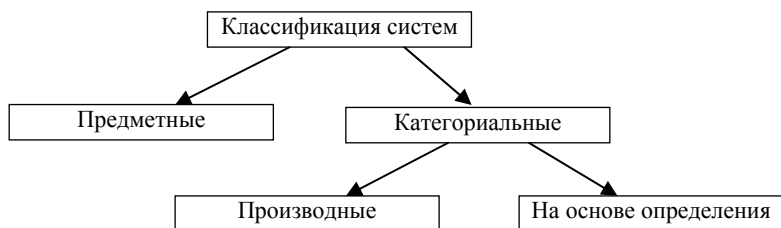


Рис. 1.3. Виды классификаций систем

В клетки данной матрицы заносятся конкретные разновидности существующих систем (обычная оконная задвижка – простая детерминированная система, а общество – вероятностная очень сложная система).

Категориальные классификации выделяют системы по некоторым признакам, общим для всех систем. Такой подход был реализован Б. А. Гладких с соавторами на основе определения системы А. И. Уемовым в триаде «вещь» – «свойство» – «отношение» (табл. 1.2).

1.1. Классификация систем по Ст. Бирю

Системы	Простые	Сложные	Очень сложные
Детерминированные			
Вероятностные			

1.2. Классификация систем по Б. А. Гладких

Категориальные характеристики	Компоненты системы		
	Свойства	Элементы	Отношения
Моно			
Поли			
Статические			
Динамические (функционирующие)			
Динамические (развивающиеся)			
Детерминистские			
Случайные			
Простые			
Сложные			

Отсюда выделяются такие типы систем:

- монофункциональная детерминистская простая (работа часового механизма, небольшого предприятия);
- монофункциональная случайная простая (те же примеры, только при наличии помех);
- монофункциональная вероятностная сложная (системы с обилием и сложностью случайных факторов);
- моноразвивающаяся детерминированная простая (предприятие, осваивающее выпуск новой продукции);
- моноразвивающаяся вероятностная простая;
- моноразвивающаяся вероятностная сложная;
- полифункциональная детерминированная простая;
- полифункциональная детерминированная сложная;
- полифункциональная вероятностная простая;
- полифункциональная вероятностная сложная;
- полиразвивающаяся детерминированная простая;
- полиразвивающаяся детерминированная сложная;
- полиразвивающаяся вероятностная простая;
- полиразвивающаяся вероятностная сложная.

Одной из распространённых является классификация С. А. Саркисяна с соавторами, в которой все системы делятся на абстрактные и материальные с последующим делением их на простые разновидности (рис. 1.4).

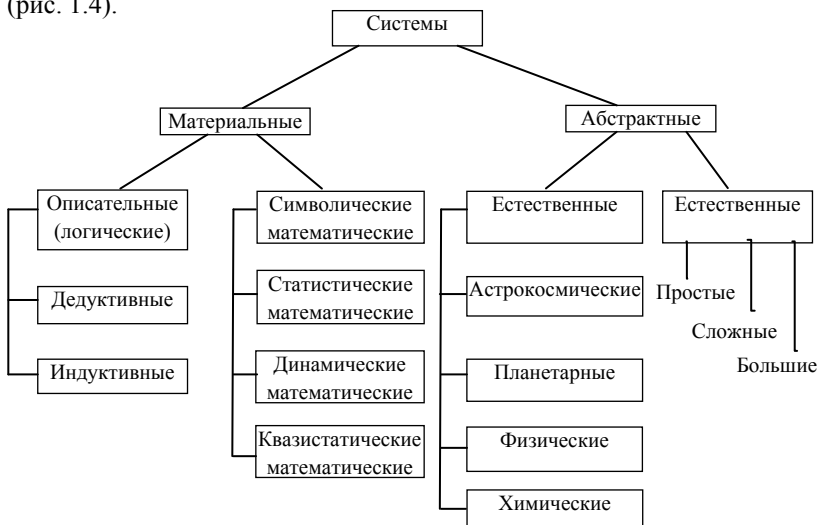


Рис. 1.4. Классификация систем по С. А. Саркисяну

Существенный недостаток данной классификации состоит в том, что из неё выпадают целые классы систем (биологические, физиологические, социальные), т.е. она не охватывает материальные системы.

Развёрнутую типологию систем дают В. В. Дружинин и Д. С. Конторов. Они разбивают их сначала на классы по природе на технические, кибернетические, биологические, общественные и интеллектуальные, а затем – на подклассы. При этом системы распадаются: технические – на простые и сложные, равновесные и неравновесные; кибернетические – на программные, адаптивные и активные, биологические – на предбиологические, простейшие, колонии специализированных организмов, колонии централизованных, колонии высших; общественные – на индивидуумы, простая группа, управляемая группа, общество, общество, большое общество, человеческое общество, сверхобщество; интеллектуальные – на конкретные и абстрактные.

Анализ классификаций показывает, что многие из них отличаются эклектичностью, несущественностью и неполнотой.

В самом общем виде можно предложить сущностную классификацию систем.

В основу любой классификации должна быть положена концепция, объясняющая классифицируемые явления. Классификация представляет собой многоступенчатое, разветвлённое деление логического объёма понятия. В результате создаётся система соподчинённых понятий: делимое понятие – род, новые понятия – виды, виды видов (подвиды) и т.д. Концептуальный подход к классификации делает её сущностной (рис. 1.5).

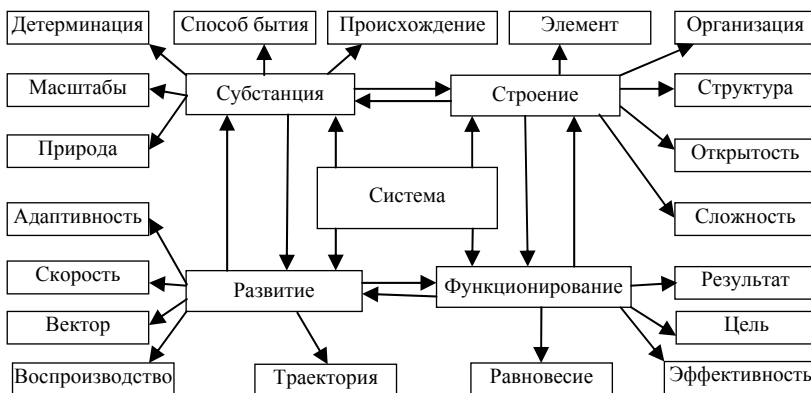


Рис. 1.5. Интерпретация основных составляющих системы

Каждая из четырёх составляющих сущностной характеристики системы может быть представлена совокупностями основополагающих параметров, соответствующих их природе. Так, субстанция может быть представлена природой систем, их сложностью, масштабами, детерминацией, происхождением и способом бытия. Для строения свойственных элементов, связи, организация, структура и сложность. Функционирование выражается равновесием, целью, результатом и эффективностью. Развитие характеризуется адаптивностью, скоростью, воспроизводством, вектором и траекторией.

Существует ряд подходов к разделению систем по сложности. В частности, Г. Н. Поваров в зависимости от числа элементов, входящих в систему, выделяет четыре класса систем:

- малые системы (10...103 элементов),
- сложные (104...107 элементов),
- ультрасложные (107...1030 элементов),
- суперсистемы (1030...10 200 элементов).

Так как понятие элемента возникает относительно задачи и цели исследования системы, то и данное определение сложности является относительным, а не абсолютным.

Английский кибернетик С. Бир классифицирует все кибернетические системы на простые и сложные в зависимости от способа описания: детерминированного или теоретико-вероятностного. А. И. Берг определяет сложную систему как систему, которую можно описать не менее чем на двух различных математических языках (например, с помощью теории дифференциальных уравнений и алгебры Буля).

Очень часто сложными системами называют системы, которые нельзя корректно описать математически либо потому, что в системе имеется очень большое число элементов, неизвестным образом связанных друг с другом, либо неизвестна природа явлений, протекающих в системе.

1.4. СВОЙСТВА (ЗАКОНОМЕРНОСТИ) СИСТЕМ

Важнейшие свойства системы: структурность, взаимозависимость со средой, иерархичность, множественность описаний (табл. 1.3).

Ограниченность системы представляет собой первое и изначальное её свойство. Это необходимое, но не достаточное свойство. Если совокупность объектов ограничена от внешнего мира, то она может быть системной, а может и не быть ею. Совокупность становится системой только тогда, когда она обретает целостность, т.е. приобретает структурность, иерархичность, взаимосвязь со средой. Система как целостность характеризуется системным способом бытия, которое

1.3. Характеристика основных свойств системы

Свойство системы	Характеристика
Ограниченность	Система отделена от окружающей среды границами
Целостность	Её свойство целого принципиально не сводится к сумме свойств составляющих элементов
Структурность	Поведение системы обусловлено не столько особенностями отдельных элементов, сколько свойствами её структуры
Взаимозависимость со средой	Система формирует и проявляет свойства в процессе взаимодействия со средой
Иерархичность	Соподчинённость элементов в системе
Множественность описаний	По причине сложности познание системы требует множественности её описаний

включает её внутреннее бытие, связанное со структурной организацией, и внешнее бытие – функционирование. Целостность, как известно, не сводима к своим составным частям. Здесь всегда наблюдается потеря качества. Поскольку научное описание объекта предполагает процедуры мысленного расчленения целостности, то целостность представляет собой некоторое множество описаний. Отсюда многообразие определений системы: структурированное множество; множество, взаимодействующее с окружением; упорядоченная целостность и т.д.

Целостность. Закономерность целостности проявляется в системе в возникновении новых интегративных качеств, не свойственных образующим её компонентам. Чтобы глубже понять закономерность целостности, необходимо рассмотреть две её стороны:

1. Свойства системы (целого) не являются суммой свойств элементов или частей (несводимость целого к простой сумме частей);
2. Свойства системы (целого) зависят от свойств элементов, частей (изменение в одной части вызывает изменение во всех остальных частях и во всей системе).

Существенным проявлением закономерности целостности являются новые взаимоотношения системы как целого со средой, отличные от взаимодействия с ней отдельных элементов.

Свойство целостности связано с целью, для выполнения которой предназначена система.

Весьма актуальным является оценка степени целостности системы при переходе из одного состояния в другое. В связи с этим возникает двойственное отношение к закономерности целостности.

Ее называют физической аддитивностью, независимостью, суммативностью, обособленностью. Свойство физической аддитивности проявляется у системы, как бы распавшейся на независимые элементы. Строго говоря, любая система находится всегда между крайними точками как бы условной шкалы: абсолютная целостность – абсолютная аддитивность, и рассматриваемый этап развития системы можно охарактеризовать степенью проявления в ней одного или другого свойства и тенденцией к его нарастанию или уменьшению.

Для оценки этих явлений А. Холл ввел такие закономерности, как «прогрессирующая факторизация» (стремление системы к состоянию со все более независимыми элементами) и «прогрессирующая систематизация» (стремление системы к уменьшению самостоятельности элементов, т.е. к большей целостности). Существуют методы введения сравнительных количественных оценок степени целостности, коэффициента использования элементов в целом с точки зрения определённой цели.

Интегративность. Этот термин часто употребляют как синоним целостности. Однако им подчёркивают интерес не к внешним факторам проявления целостности, а к более глубоким причинам формирования этого свойства и, главное, к его сохранению. Интегративными называют системообразующие, системоохраняющие факторы, важными среди которых являются неоднородность и противоречивость её элементов.

Коммуникативность. Эта закономерность составляет основу определения системы, предложенного В. Н. Садовским и Э. Г. Юдиным в книге «Исследования по общей теории систем». Система образует особое единство со средой; как правило, любая исследуемая система представляет собой элемент системы более высокого порядка; элементы любой исследуемой системы, в свою очередь, обычно выступают как системы более низкого порядка.

Иными словами, система не изолирована, она связана множеством коммуникаций со средой, которая не однородна, а представляет собой сложное образование, содержит надсистему (или даже надсистемы), задающую требования и ограничения исследуемой системе, подсистемы и системы одного уровня с рассматриваемой.

Рассмотрим иерархичность как закономерность построения всего мира и любой выделенной из него системы. Иерархическая упорядо-

ченность пронизывает всё, начиная от атомно-молекулярного уровня и кончая человеческим обществом. Иерархичность как закономерность заключается в том, что закономерность целостности проявляется на каждом уровне иерархии. Благодаря этому на каждом уровне возникают новые свойства, которые не могут быть выведены как сумма свойств элементов. При этом важно, что не только объединение элементов в каждом узле приводит к появлению новых свойств, которых у них не было, и утрате некоторых свойств элементов, но и что каждый член иерархии приобретает новые свойства, отсутствующие у него в изолированном состоянии.

Таким образом, на каждом уровне иерархии происходят сложные качественные изменения, которые не всегда могут быть представлены и объяснены. Но именно благодаря этой особенности рассматриваемая закономерность приводит к интересным следствиям. Во-первых, с помощью иерархических представлений можно отображать системы с неопределённостью.

Во-вторых, построение иерархической структуры зависит от цели: для многоцелевых ситуаций можно построить несколько иерархических структур, соответствующих разным условиям, и при этом в разных структурах могут принимать участие одни и те же компоненты. В-третьих, даже при одной и той же цели, если поручить формирование иерархической структуры разным исследователям, то в зависимости от их предшествующего опыта, квалификации и знания системы они могут получить разные иерархические структуры, т.е. по-разному разрешить качественные изменения на каждом уровне иерархии.

Эквифинальность – это одна из наименее исследованных закономерностей. Она характеризует предельные возможности систем определённого класса сложности. Л. фон Берталанфи, предложивший этот термин, определяет эквифинальность применительно к «открытой» системе как способность (в отличие от состояний равновесия в закрытых системах) полностью детерминированных начальными условиями систем достигать не зависящего от времени состояния (которое не зависит от её исходных условий и определяется исключительно параметрами системы). Потребность во введении этого понятия возникает, начиная с некоторого уровня сложности, например биологические системы.

В настоящее время не исследован ряд вопросов этой закономерности: какие именно параметры в конкретных системах обеспечивают свойство эквивалентности? как обеспечивается это свойство? как проявляется закономерность эквивалентности в организационных системах?

Историчность. Время является непременной характеристикой системы, поэтому каждая система исторична, и это такая же закономерность, как целостность, интегративность и др. Легко привести

примеры становления, расцвета, упадка и даже смерти биологических и общественных систем, но для технических и организационных систем определить периоды развития довольно трудно. Основа закономерности историчности – внутренние противоречия между компонентами системы. Но как управлять развитием или хотя бы понимать приближение соответствующего периода развития системы – эти вопросы еще мало исследованы.

В последнее время на необходимость учёта закономерности историчности начинают обращать больше внимания. В частности, в системотехнике при создании сложных технических комплексов требуется на стадии проектирования системы рассматривать не только вопросы разработки и обеспечения развития системы, но и вопрос, как и когда нужно её уничтожить. Например, списание техники, особенно сложной – авиационной, захоронение ядерных установок и др.

Закон необходимого разнообразия. Его впервые сформулировал У. Р. Эшби: чтобы создать систему, способную справиться с решением проблемы, обладающей определённым, известным разнообразием, нужно, чтобы сама система имела ещё большее разнообразие, чем разнообразие решаемой проблемы, или была способна создать в себе это разнообразие. Этот закон достаточно широко применяется на практике. Он позволяет, например, получить рекомендации по совершенствованию системы управления предприятием, объединением, отраслью.

Закономерность осуществимости и потенциальной эффективности систем. Исследования взаимосвязи сложности структуры системы со сложностью её поведения позволили получить количественные выражения предельных законов для таких качеств системы, как надёжность, помехоустойчивость, управляемость и др. На основе этих законов оказалось возможным получение количественных оценок порогов осуществимости систем с точки зрения того или иного качества, а объединяя качества – предельные оценки жизнеспособности и потенциальной эффективности сложных систем.

Закономерность целеобразования. Исследования процесса целеобразования в сложных системах философами, психологами и кибернетиками позволили сформулировать некоторые общие закономерности процессов обоснования и структуризации целей в конкретных условиях совершенствования сложных систем.

Зависимость представления о цели и формулировки цели от стадии познания объекта (процесса). Анализ понятия «цель» позволяет сделать вывод, что, формулируя цель, нужно стремиться отразить в формулировке или в способе представления цели её активную роль в познании и в то же время сделать её реалистичной, направить с её помощью деятельность на получение определённого результата. При

этом формулировка цели и представление о ней зависит от стадии познания объекта и в процессе развития представления об объекте цель может переформулироваться. Коллектив, формирующий цель, должен определить, в каком смысле на данном этапе рассмотрения объекта употребляется понятие цели, к какой точке «условной шкалы» («идеальное устремление в будущее» – «конкретный результат деятельности») ближе принимаемая формулировка цели.

Зависимость цели от внутренних и внешних факторов. При анализе причин возникновения цели нужно учитывать как внешние по отношению к выделенной системе факторы (внешние потребности, мотивы, программы), так и внутренние потребности, мотивы, программы («самодвижение» целостности). При этом цели могут возникать на основе противоречий как между внешними и внутренними факторами, так и между внутренними факторами, имевшимися ранее и вновь возникающими в находившейся в постоянном самодвижении целостности. Это очень важное отличие организационных «развивающихся», открытых систем от технических (замкнутых, закрытых) систем. Теория управления техническими системами оперирует понятием цели только по отношению к внешним факторам, а в открытых, развивающихся системах цель формируется внутри системы, и внутренние факторы, влияющие на формирование целей, являются такими же объективными, как и внешние.

Возможность сведения задачи формирования общей (главной, глобальной) цели к задаче структуризации цели. Анализ процессов формулирования глобальной цели в сложной системе показывает, что эта цель возникает в сознании руководителя или коллектива не как единичное понятие, а как некоторая, достаточно «размытая» область. На любом уровне цель возникает вначале в виде «образа» цели. При этом достичь одинакового понимания общей цели всеми исполнителями, по-видимому, принципиально невозможно без её детализации в виде упорядоченного или неупорядоченного набора взаимосвязанных подцелей, которые делают её понятной и более конкретной для разных исполнителей. Таким образом, задача формулирования общей цели в сложных системах должна быть сведена к задаче структуризации цели.

Следующие закономерности являются продолжением двух первых применительно к структурам цели.

Зависимость способа представления структуры целей от стадии познания объекта или процесса (продолжение первой закономерности). Наиболее распространённым способом представления структур целей является древовидная иерархическая структура. Существуют и другие способы отображения: иерархия со «слабыми» связями, табличное или матричное представление, сетевая модель. Иерархическое

и матричное описание – это декомпозиция цели в пространстве, сетевая модель – декомпозиция во времени. Промежуточные подцели могут формулироваться по мере достижения предыдущей, что может использоваться как средство управления. Перспективным представляется развёртывание иерархических структур во времени, т.е. сочетание декомпозиции цели в пространстве и во времени.

Проявление в структуре целей закономерности целостности. В иерархической структуре целей, как и в любой иерархической структуре, закономерность целостности проявляется на каждом уровне иерархии. Применительно к структуре целей это означает, что достижение целей вышележащего уровня не может быть полностью обеспечено достижением подцелей, хотя и зависит от них, и что потребности, мотивы, программы, влияющие на формирование целей, нужно исследовать на каждом уровне иерархии.

1.5. ОСОБЕННОСТИ СИСТЕМНОГО ПОДХОДА И СИСТЕМНОГО АНАЛИЗА

Системный подход разработан в середине XX века и связан с решением проблем в лабораториях телефонии Белла.

С общефилософской точки зрения это метод познания объекта как системы. Он заключается в анализе свойства целостности связей между подсистемами и элементами.

Применение системных представлений для анализа сложных объектов и процессов рассматривают системные направления, включающие в себя: системный подход, системные исследования, системный анализ (системологию, системотехнику и т.п.). За исключением системотехники, область которой ограничена техническими системами, все другие термины часто употребляются как синонимы. Однако в последнее время системные направления начали применять в более точном смысле.

Системный подход. Этот термин начал применяться в первых работах, в которых элементы общей теории систем использовались для практических приложений. Используя этот термин, подчёркивали необходимость исследования объекта с разных сторон, комплексно, в отличие от ранее принятого разделения исследований на физические, химические и др. Оказалось, что с помощью многоаспектных исследований можно получить более правильное представление о реальных объектах, выявить их новые свойства, лучше определить взаимоотношения объекта с внешней средой, другими объектами. Заимствованные при этом понятия теории систем вводились не строго, не исследовался вопрос, каким классом систем лучше отобразить объект, какие

свойства и закономерности этого класса следует учитывать при конкретных исследованиях и т.п. Иными словами, термин «системный подход» практически использовался вместо терминов «комплексный подход», «комплексные исследования».

Системные исследования. В работах под этим названием понятия теории систем используются более конструктивно: определяется класс систем, вводится понятие структуры, а иногда и правила её формирования и т.п. Это был следующий шаг в системных направлениях. В поисках конструктивных рекомендаций появились системные направления с разными названиями: системотехника, системология и др. Для их обобщения стал применяться термин «системные исследования». Часто в работах использовался аппарат исследования операций, который к тому времени был больше развит, чем методы конкретных системных исследований.

Системный анализ. В настоящее время системный анализ является наиболее конструктивным направлением. Этот термин применяется неоднозначно. В одних источниках он определяется как «приложение системных концепций к функциям управления, связанным с планированием». В других – как синоним термина «анализ систем» или термина «системные исследования». Однако независимо от того, применяется он только к определению структуры целей системы, к планированию или к исследованию системы в целом, включая и функциональную и обеспечивающую части, работы по системному анализу существенно отличаются от рассмотренных выше тем, что в них всегда предлагается методология проведения исследований, делается попытка выделить этапы исследования и предложить методику выполнения этих этапов в конкретных условиях. В этих работах всегда уделяется особое внимание определению целей системы, вопросам формализации представления целей. Некоторые авторы даже подчёркивают это в определении: системный анализ – это методология исследования целенаправленных систем (Д. Киланд, В. Кинг).

Термин «системный анализ» впервые появился в связи с задачами военного управления в исследованиях RAND Corporation (1948), а в отечественной литературе получил широкое распространение после выхода в 1969 г. книги С. Оптнера «Системный анализ для решения деловых и промышленных проблем».

Вначале работы по системному анализу в большинстве случаев базировались на идеях теории оптимизации и исследования операций. При этом особое внимание уделялось стремлению в той или иной форме получить выражение, связывающее цель со средствами, аналогичное критерию функционирования или показателю эффективности, т.е. отобразить объект в виде хорошо организованной системы.

Так, например, в ранних руководящих материалах по разработке автоматизированных систем управления (АСУ) рекомендовалось цели представлять в виде набора задач и составлять матрицы, связывающие задачи с методами и средствами достижения. Правда, при практическом применении этого подхода довольно быстро выяснялась его недостаточность, и исследователи стали прежде всего обращать внимание на необходимость построения моделей, не просто фиксирующих цели, компоненты, связи между ними, а позволяющих накапливать информацию, вводить новые компоненты, выявлять новые связи и т.д., т.е. отображать объект в виде развивающейся системы, не всегда предлагая, как это делать.

Позднее системный анализ начинают определять как «процесс последовательного разбиения изучаемого процесса на подпроцессы» (С. Янг) и основное внимание уделяют поиску приёмов, позволяющих организовать решение сложной проблемы путём расчленения её на подпроблемы и этапы, для которых становится возможным подобрать методы исследования и исполнителей. В большинстве работ стремились представить многоступенчатое расчленение в виде иерархических структур типа «дерева», но в ряде случаев разрабатывались методики получения вариантов структур, определяемых временными последовательностями функций.

В настоящее время системный анализ развивается применительно к проблемам планирования и управления, и в связи с усилением внимания к программно-целевым принципам в планировании этот термин стал практически неотделим от терминов «целеобразование» и «программно-целевое планирование и управление». В работах этого периода системы анализируются как целое, рассматриваются роль процессов целеобразования в развитии целого, роль человека. При этом оказалось, что в системном анализе не хватает средств: развиты в основном средства расчленения на части, но почти нет рекомендаций, как при расчленении не утратить целое. Поэтому наблюдается усиление внимания к роли неформализованных методов при проведении системного анализа. Вопросы сочетания и взаимодействия формальных и неформальных методов при проведении системного анализа не решены. Но развитие этого научного направления идёт по пути их решения.

В системном анализе при появлении работ Янга стали уделять внимание разбиению различного рода изучаемых объектов или процессов на элементы или подпроцессы, что приобрело название декомпозиции.

Методологические подходы в системном анализе объединяют совокупность сложившихся в практике аналитической деятельности приёмов и способов реализации системной деятельности. Их характеристика представлена в табл. 1.4.

1.4. Характеристика основных подходов в системном анализе

Подходы в системном анализе	Характеристика подходов
Системный	<p>Несводимость свойств целого к сумме свойств элементов.</p> <p>Поведение системы определяется как особенностями отдельных элементов, так и особенностями её структуры.</p> <p>Существует зависимость между внутренними и внешними функциями системы.</p> <p>Система находится во взаимодействии с внешней средой, обладает соответствующей ей внутренней средой.</p> <p>Система представляет собой развивающуюся целостность</p>
Структурно-функциональный	<p>Выявление структуры (или функций) системы.</p> <p>Установление зависимости между структурой и функциями системы. Построение соответственно функций (или структуры) системы</p>
Конструктивный	<p>Реалистический анализ проблемы. Анализ всех возможных вариантов разрешения проблемы. Конструирование системы, действие по разрешению проблемы</p>
Комплексный	<p>Изучение всех сторон, свойств, многообразия структур, функций системы, её связей со средой. Рассмотрение их в единстве. Выяснение степени значимости взятых в единстве характеристик системы в её сущности</p>
Проблемный	<p>Выделение проблемы как противоречия между какими-либо сторонами объекта, определяющими его развитие.</p> <p>Определение типа проблемы, её оценка.</p> <p>Выработка способов разрешения проблемы</p>

Подходы в системном анализе	Характеристика подходов
Ситуационный	<p>Выделение проблемного комплекса, лежащего в основе ситуации. Выделение основных характеристик ситуации.</p> <p>Установление причин возникновения ситуаций и следствий их развёртывания.</p> <p>Оценка ситуации, её прогнозирование.</p> <p>Разработка программы деятельности в данной ситуации</p>
Инновационный	<p>Констатация проблемы обновления.</p> <p>Формирование модели нововведения, обеспечивающего разрешение проблемы. Внедрение нововведения.</p> <p>Управление нововведением, его освоение и реализация</p>
Нормативный	<p>Констатация проблемы системы.</p> <p>Установление рациональных норм системы.</p> <p>Преобразование системы в соответствии с нормами</p>
Целевой	<p>Определение цели системы.</p> <p>Декомпозиция цели на простые составляющие.</p> <p>Обоснование целей. Построение «дерева целей».</p> <p>Оценка экспертами всех «ветвей» «дерева целей» относительно времени и ресурсов достижения</p>
Деятельностный	<p>Определение проблемы.</p> <p>Определение объекта деятельности.</p> <p>Формулировка целей и задач деятельности.</p> <p>Определение субъекта деятельности.</p> <p>Формирование модели деятельности.</p> <p>Осуществление деятельности</p>

Подходы в системном анализе	Характеристика подходов
Морфологический	<p>Максимально точное определение проблемы.</p> <p>Нахождение наибольшего числа в пределах всех возможных вариантов разрешения проблемы.</p> <p>Реализация системы путём комбинирования основных структурных элементов или признаков.</p> <p>Применение методов морфологического моделирования: систематического покрытия поля; отрицания и конструирования; морфологического ящика; сопоставления совершенного с дефектным, обобщения и др.</p>
Программно-целевой	<p>Определение проблемы.</p> <p>Формулирование целей.</p> <p>Построение программы достижения целей</p>

Важнейшей, если не главной составной частью методологии системного анализа выступают *методы*, вариант классификации которых представлен в табл. 1.5. В качестве оснований классификации используется тип знания, обрабатываемый методом; способ реализации, в качестве которого могут выступать либо интуиция, либо знание; выполняемые функции, сводящиеся к получению, представлению и обработке информации; уровень знания – теоретический либо эмпирический; форма представления знания, которая может быть качественной либо количественной.

Методологический комплекс системного анализа был бы неполным, если в нём не выделить его теоретический ансамбль. Теория является не только отражением действительности, но и методом её отражения, т.е. она выполняет методологическую функцию. На этом основании системные теории включаются в системный методологический комплекс. Наиболее важные системные теории, которые воздействуют на анализ, представлены в табл. 1.6.

1.5. Методы системного анализа

Основание классификации	Методы системного анализа
Тип знания	<p>Философские методы (диалектический, метафизический и т.п.).</p> <p>Общенаучные методы (системный, структурно-функциональный, моделирование, формализация и т.п.).</p> <p>Частно научные методы (свойственны конкретной науке: методы моделирования социальных, биологических систем и т.п.).</p> <p>Дисциплинарные методы (применяются в той или иной дисциплине, входящей в какую-нибудь отрасль науки, семиотические, лингвистические и т.п.)</p>
Способ реализации	<p>Интуитивные методы («мозговая атака», «сценарии», экспертные методы и т.п.).</p> <p>Научные методы (анализ, классификация, системного моделирования, методы логики и теории множеств и т.п.)</p>
Выполняемые функции	<p>Методы получения информации (системное наблюдение, описание, экспертные методы, игровые методы и т.п.).</p> <p>Методы представления информации (группировка, классификация и т.п.).</p> <p>Методы анализа информации (классификация, обобщение, методы анализа информационных систем и т.п.)</p>
Уровень знания	<p>Теоретические методы (анализ, синтез, теоретизация и т.п.).</p> <p>Эмпирические методы (игровые методы, морфологические методы, экспертные оценки и т.п.)</p>
Форма представления знания	<p>Качественные методы, опирающиеся на качественный подход к объекту (метод «сценариев», морфологические методы).</p> <p>Количественные методы, использующие аппарат математики (метод «Дельфи», статистические методы, методы теории графов, комбинаторики, кибернетики, логики, теории множеств, лингвистики, исследования операций, семиотики, топологии и т.п.)</p>

1.6. Системные теории, их авторы и характеристика

Название	Автор	Характеристика
Общая теория систем (несколько вариантов)	А. А. Богданов, Л. Бергаланфи, М. Месарович, У. Росс Эшби, А. И. Уемов, В. С. Тютин, Ю. А. Урманцев и др.	Формирование понятийного аппарата систем. Попытка создания строгой теории. Выявление общих закономерностей функционирования и развития систем любой природы
Структурализм (несколько вариантов)	К. Леви-Стросс, М. П. Фуко, Ж. Лакан, Р. Барт, Л. Гольдман, А. Р. Радклифф-Браун и др.	Выявление структур, имеющих в культуре. Применение структурных методов в изучении различных продуктов человеческой деятельности в целях выявления логики порождения, строения и функционирования объектов духовной культуры. Выделение и анализ эпистем – способов фиксации связей между словами и вещами
Функционализм (несколько вариантов)	Г. Спенсер, Т. Парсонс, Б. Малиновский, Р. Мертон, Н. Луман, К. Гемпель, Ч. Миллс и др.	Выявление функций как наблюдаемых следствий, которые служат саморегуляции и адаптации системы. Исследование функциональных потребностей и их обеспечения структурами. Выделение явных и латентных функций, функций и дисфункций. Исследование проблем адаптации и саморегуляции систем
Структурный функционализм (несколько вариантов)	Р. Бейлз, Р. Мак-Айвера, Р. Мертон, Т. Парсонс, Н. Смелсер, Э. Шилз и др.	Равновесие и спонтанная регуляция систем. Наличие в обществе инструментальной и функциональной рациональности. Общество как система имеет технико-экономическую, профессиональную и стратификационную структуры

Название	Автор	Характеристика
Системно-кибернетические теории	Н. Винер, У. Росс Эшби, Р. Акофф, Ст. Бир, В. М. Глушков и др.	Выделение общих законов управления. Гомеостатический, целевой, управленческий характер систем. Наличие прямой и обратной отрицательной и положительной обратной связей. Процессы управления рассматриваются как процессы переработки информации. Теория автоматического регулирования. Теория информации. Теория оптимального управления. Теория алгоритмов
Математические теории систем (несколько вариантов)	М. Месарович, Л. В. Кантарович, В. С. Немчинов и др.	Математические определения систем, основанные на теории множеств, логике, математическом программировании, теории вероятностей и статистике. Математические описания структуры, функций и состояний систем
Синергетика	И. И. Пригожин, Г. Хаген	Исследование процессов самоорганизации в системах любой природы. Объяснение поведения сложных нелинейных систем, находящихся в неравновесных состояниях, спонтанным образованием структур. Роль динамического хаоса и флуктуаций в развитии системы. Наличие многообразия путей развития систем в условиях хаоса

1.6. СИСТЕМНЫЕ ПОНЯТИЯ ИНФОРМАЦИОННОГО ПРОЦЕССА, ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ, ИНФОРМАЦИОННОЙ СИСТЕМЫ

Теория информационных процессов (ТИП) – совокупность моделей и методов, предназначенных (или используемых) для анализа информационных процессов (ИП), происходящих в технических, экономических, социальных, биологических, экологических и других системах.

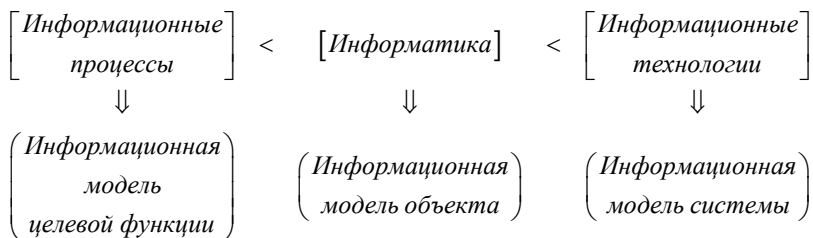
Основные информационные процессы: формирование, преобразование, сбор, хранение, обработка, распределение, использование информации и др. Каждый информационный процесс характеризуется *целевой функцией*, реализуемой конкретными техническими средствами.

Каждой целевой функции может быть сопоставлена *информационная модель*, описываемая математическими выражениями, условными обозначениями или символами (информационно-математическая модель целевой функции).

Информатика – область науки и техники, изучающая информационные процессы и методы их автоматизации средствами вычислительной техники. Информатика как наука формирует методологические основы построения информационной модели объекта. Создание такой модели (для организации целенаправленной деятельности в технических, биологических, экономических, социальных и других системах) осуществляется на основе реализации информационных процессов.

Информационные технологии, в широком понимании, – это разработка алгоритмов, программ и их применение на компьютере с целью анализа и синтеза сложных систем (технических, информационных, биологических, экономических, социальных и др.).

Таким образом, соотношение понятий «информационные процессы», «информатика» и «информационные технологии» может быть выражено следующим образом ([...] – физический уровень представления, (...) – информационный уровень представления):



Согласно определению, принятому ЮНЕСКО, *информационные технологии* – это комплекс взаимосвязанных, научных, технологических, инженерных дисциплин, изучающих методы эффективной орга-

низации труда людей, занятых обработкой и хранением информации; вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы.

Основная цель ИТ – в результате целенаправленных действий по переработке первичной информации получить необходимую для пользователя информацию с целью её анализа и принятия на его основе решения по выполнению какого-либо действия.

Информационная система (ИС) – это организационно-упорядоченная взаимосвязанная совокупность средств и методов ИТ, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. Такое понимание информационной системы предполагает использование в качестве основного технического средства переработки информации ЭВМ и средств телекоммуникаций, реализующих информационные процессы и выдачу информации, необходимой в процессе принятия решений задач из любой области.

Реализация функций ИС невозможна без знания ориентированной на неё ИТ, а с другой стороны, ИТ может существовать и вне сферы ИС. Таким образом, ИТ является более ёмким понятием, отражающим современное представление о процессах преобразования информации в информационном обществе.

В зависимости от конкретной области применения ИС могут очень сильно различаться по своим функциям, архитектуре, реализации. Можно выделить основные свойства, которые являются общими для всех ИС:

- структура ИС, её функциональное назначение должны соответствовать поставленным целям;
- производство достоверной, надёжной, своевременной и систематизированной информации, основанной на использовании БД, экспертных систем и баз знаний. Так как любая ИС предназначена для сбора, хранения и обработки информации, то в основе любой ИС лежит среда хранения и доступа к данным. Среда должна обеспечивать уровень надёжности хранения и эффективность доступа, которые соответствуют области применения ИС;
- ИС должна контролироваться людьми, ими пониматься и использоваться в соответствии с основными принципами, реализованными в виде стандарта предприятия на ИС. Интерфейс пользователя ИС должен быть легко понимаем на интуитивном уровне;
- использование сетей передачи данных.

ИС решают следующие основные задачи:

– *Поиск, обработка и хранение информации*, которая долго накапливается и утрата которой невозможна. Компьютеризованные ИС предназначены для более быстрой и надёжной обработки информации, чтобы люди не тратили время и избежали свойственных человеку случайных ошибок, чтобы сэкономить расходы и сделать жизнь людей более комфортной.

– *Хранение данных разной структуры*. Не существует развитой ИС, работающей с одним однородным файлом данных. Более того, разумным требованием к информационной системе является то, чтобы она могла развиваться. Могут появиться новые функции, для выполнения которых требуются дополнительные данные с новой структурой. При этом вся накопленная ранее информация должна остаться сохранной.

– *Анализ и прогнозирование потоков информации различных видов и типов, перемещающихся в обществе*. Изучаются потоки с целью их минимизации, стандартизации и приспособления для эффективной обработки на вычислительных машинах, а также особенности потоков информации, протекающей через различные каналы распространения информации.

– *Исследование способов представления и хранения информации*, создание специальных языков для формального описания информации различной природы, разработка специальных приёмов сжатия и кодирования информации, аннотирования объёмных документов и реферирования их. В рамках этого направления развиваются работы по созданию банков данных большого объёма, хранящих информацию из различных областей знаний в форме, доступной для вычислительных машин.

– *Построение процедур и технических средств для их реализации*, с помощью которых можно автоматизировать процесс извлечения информации из документов, не предназначенных для вычислительных машин, а ориентированных на восприятие их человеком.

– *Создание информационно-поисковых систем*, способных воспринимать запросы к информационным хранилищам, сформулированные на естественном языке, а также специальных языках запросов для систем такого типа.

– *Создание сетей хранения, обработки и передачи информации*, в состав которых входят информационные банки данных, терминалы, обрабатывающие центры и средства связи.

Конкретные задачи, которые должны решаться информационной системой, зависят от той прикладной области, для которой предназначена система.

Тенденции развития современных ИТ приводят к постоянному возрастанию сложности ИС, создаваемых в различных областях. Современные крупные проекты ИС характеризуются, как правило, несколькими особенностями:

– *Сложность описания* – наличие достаточно большого количества функций, процессов, элементов данных и сложные взаимосвязи между ними, требующие тщательного моделирования и анализа данных и процессов.

– *Наличие совокупности тесно взаимодействующих компонентов (подсистем)*, имеющих свои локальные задачи и цели функционирования (например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, и приложений аналитической обработки (поддержки принятия решений), использующих нерегламентированные запросы к данным большого объема).

– *Отсутствие прямых аналогов*, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем.

– *Необходимые интеграции* существующих и вновь разрабатываемых приложений.

– *Функционирование* в неоднородной среде, на нескольких аппаратных платформах.

– *Разобщённость и разнородность* отдельных групп разработчиков по уровню квалификации и сложившимся традициям использования тех или иных инструментальных средств.

– *Существенная временная протяжённость проекта*, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков и, с другой стороны, масштабами организации заказчика и различной степенью готовности отдельных её подразделений к внедрению ИС.

Задания для самопроверки

1. Дайте определения модели системы.
2. Приведите пример классификации систем.
3. Перечислите важнейшие свойства систем.
4. Назовите методы системного анализа.
5. Перечислите основные свойства информационных систем.

2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

2.1. ОБЩИЕ ПОДХОДЫ К МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ СИСТЕМ

При создании и эксплуатации сложных систем требуется проводить многочисленные исследования и расчёты, связанные с оценкой показателей, характеризующих различные свойства систем; выбором оптимальной структуры системы; выбором оптимальных значений её параметров.

Выполнение таких исследований возможно лишь при наличии математического описания процесса функционирования системы, т.е. её математической модели.

Сложность реальных систем не позволяет строить для них «абсолютно» адекватные модели. Математическая модель (ММ) описывает некоторый упрощённый процесс, в котором представлены лишь основные явления, входящие в реальный процесс, и лишь главные факторы, действующие на реальную систему.

Какие явления считать основными и какие факторы главными – существенно зависит от назначения модели, от того, какие исследования с её помощью предполагается проводить. Поэтому процесс функционирования одного и того же реального объекта может получить различные математические описания в зависимости от поставленной задачи.

Так как ММ сложной системы может быть сколько угодно много и все они определяются принятым уровнем абстрагирования, то рассмотрение задач на каком-либо одном уровне абстракции позволяет дать ответы на определённую группу вопросов, а для получения ответов на другие вопросы необходимо провести исследование уже на другом уровне абстракции. Каждый из возможных уровней абстрагирования обладает ограниченными, присущими только данному уровню абстрагирования возможностями. Для достижения максимально возможной полноты сведений необходимо изучить одну и ту же систему на всех целесообразных для данного случая уровнях абстракции.

Наиболее пригодными являются следующие уровни абстрактного описания систем: символический, или, иначе, лингвистический; теоретико-множественный; абстрактно-алгебраический; топологический; логико-математический; теоретико-информационный; динамический; эвристический.

Условно первые четыре уровня относятся к высшим уровням описания систем, а последние четыре – к низшим.

Рассмотрим подробнее уровни описания систем.

Лингвистический уровень описания – наиболее высокий уровень абстрагирования. Из него как частные случаи можно получить другие уровни абстрактного описания систем более низкого ранга. Формальные построения наиболее успешно используются, когда удастся с предметами или процессами действительности каким-то образом сопоставлять некоторые стабильные, неизменные понятия.

Понятие «высказывание» на данном абстрактном языке означает, что имеется некоторое предложение (формула), построенное на правилах данного языка. Предполагается, что эта формула содержит варьируемые переменные, которые только при определённом их значении делают высказывание истинным. Все высказывания делят обычно на два типа. К первому причисляют термы (имена предметов, члены предложения и т.д.) – высказывания, с помощью которых обозначают объекты исследования, а ко второму – функторы – высказывания, определяющие отношения между термами.

С помощью термов и функторов можно показать, как из лингвистического уровня абстрактного описания (уровня высшего ранга) как частный случай возникает теоретико-множественный уровень абстрагирования (уровень более низкого ранга).

Термы – некоторые множества, с помощью которых перечисляют элементы, или, иначе, подсистемы изучаемых систем, а функторы устанавливают характер отношений между введёнными множествами. Множество образуется из элементов, обладающих некоторыми свойствами и находящихся в некоторых отношениях между собой и элементами других множеств. Следовательно, автоматизированные системы управления (АСУ) вполне подходят под такого рода определение понятия «множество». Это доказывает, что построение сложных систем на теоретико-множественном уровне абстракции вполне уместно и целесообразно.

На *теоретико-множественном* уровне абстракции можно получить только общие сведения о реальных системах, а для более конкретных целей необходимы другие абстрактные модели, которые позволили бы производить более тонкий анализ различных свойств реальных систем. Эти более низкие уровни абстрагирования, в свою очередь, являются уже частными случаями по отношению к теоретико-множественному уровню формального описания систем.

Так, если связи между элементами рассматриваемых множеств устанавливаются с помощью некоторых однозначных функций, отображающих элементы множества в само исходное множество, то приходим к *абстрактно-алгебраическому* уровню описания систем. В таких случаях говорят, что между элементами множеств установлены

нульарные (никакие, отсутствующие), унарные, бинарные (двойные, двойственные), тернарные отношения и т.д. Если же на элементах рассматриваемых множеств определены некоторые топологические структуры, то в этом случае приходим к *топологическому* уровню абстрактного описания систем. При этом может быть использован язык общей топологии или её ветвей, именуемых гомологической топологией, алгебраической топологией и т.д.

Низшие уровни описания систем. *Логико-математический* уровень описания систем нашёл широкое применение для: формализации функционирования автоматов; задания условий функционирования автоматов; изучения вычислительной способности автоматов.

Понятие «автомат» (от греч. *automatos* – самодействующий) имеет следующие значения:

1) устройство, выполняющее некоторый процесс без непосредственного участия человека. В глубокой древности это часы, механические игрушки; со второй половины XVIII в. широкое применение автоматы нашли в промышленности для замены физического труда человека; в 40 – 50-х годах XX в. появились автоматы для выполнения некоторых видов умственного труда, автоматические вычислительные машины и другие кибернетические устройства. Применение автоматов значительно повышает производительность труда, скорость и точность выполнения операций. Освобождает человека от утомительного однообразного труда, защищает человека от условий, опасных для жизни или вредных для здоровья. Автоматы используются там, где невозможно присутствие человека (высокая температура, давление, ускорение, вакуум и т.д.);

2) математическое понятие, математическая модель реальных (технических) автоматов. Абстрактно автомат можно представить как некоторое устройство («черный ящик»), имеющее конечное число входных и выходных каналов и некоторое множество внутренних состояний. На входные каналы извне поступают сигналы, и в зависимости от их значения и своего состояния автомат переходит в следующее состояние и выдаёт сигналы на свои выходные каналы. С течением времени входные сигналы изменяются, соответственно изменяются и состояние автомата, и его выходные каналы. Таким образом, автомат функционирует во времени;

3) в узком смысле автомат употребляется для обозначения так называемых синхронных дискретных автоматов. Такие автоматы имеют конечные множества значений входных и выходных сигналов, называемых входным и выходным алфавитом. Время разбивается на промежутки одинаковой длительности (такты): на протяжении всего такта входной сигнал, состояние и выходной сигнал не изменяются.

Изменения происходят только на границах тактов. Следовательно, время можно считать дискретным $t = 1, 2, \dots, n$.

При любом процессе управления или регулирования, осуществляемом живым организмом или автоматически действующей машиной либо устройством, происходит переработка входной информации в выходную. Поэтому при *теоретико-информационном* уровне абстрактного описания систем информация выступает как свойство объектов и явлений (процессов) порождать многообразие состояний, которые посредством отражения передаются от одного объекта к другому и запечатлеваются в его структуре (возможно, в изменённом виде).

Отображение множества состояний источника во множество состояний носителя информации называется способом кодирования, а образ состояния при выбранном способе кодирования – кодом этого состояния.

Абстрагируясь от физической сущности носителей информации и рассматривая их как элементы некоторого абстрактного множества, а способ их расположения как отношение в этом множестве, приходят к абстрактному понятию кода информации как способа её представления. При таком подходе код информации можно рассматривать как математическую модель, т.е. абстрактное множество с заданными на нём предикатами. Эти предикаты определяют тип элементов кода и расположение их друг относительно друга.

Предикат – одно из фундаментальных понятий математики – условие, сформулированное в терминах точного логико-математического языка. Предикат содержит обозначения для произвольных объектов некоторого класса (переменные). При замещении переменных именами объектов данного класса предикат задаёт точно определённое высказывание.

Динамический уровень абстрактного описания систем связан с представлением системы как некоторого объекта, куда в определённые моменты времени можно вводить вещество, энергию и информацию, а в другие моменты времени – выводить их, т.е. динамическая система наделяется свойством иметь «входы» и «выходы», причём процессы в них могут протекать как непрерывно, так и в дискретные моменты времени. Кроме этого, для динамических систем вводится понятие «состояние системы», характеризующее её внутреннее свойство.

Эвристический уровень абстрактного описания систем предусматривает поиски удовлетворительного решения задач управления в связи с наличием в сложной системе человека. Эврика – это догадка, основанная на общем опыте решения родственных задач. Изучение интеллектуальной деятельности человека в процессе управления имеет очень важное значение.

Эвристика вообще – это приём, позволяющий сокращать количество просматриваемых вариантов при поиске решения задачи. Причём этот прием не гарантирует наилучшее решение.

Например, человек, играя в шахматы, пользуется эвристическими приёмами выработки решения, так как продумать весь ход игры с начала до конца практически невозможно из-за слишком большого числа вариантов игры (надо обдумать около 10^{120} вариантов). Если на один вариант затрачивать всего 10 с, а в году около $3 \cdot 10^7$ с, то при 8-часовой работе без выходных дней и отпуска человек способен просчитать в год не более $(1/3 \cdot 3 \cdot 10^7)/10 = 10^6$ вариантов. Следовательно, на перебор всех возможных вариантов шахматной партии понадобится одному человеку 10^{14} лет.

Поэтому в настоящее время бурно развивается эвристическое программирование – программирование игровых ситуаций, доказательства теорем, перевода с одного языка на другой, дифференциальной диагностики, распознавания образов (звуковых, зрительных и т.д.).

Большое внимание сейчас уделяется созданию искусственного и гибридного интеллекта. При этом, важное значение играют решения проблемы иерархически организованного перебора, создание и разработка методов отсечения заведомо невыгодных путей.

Таким образом, обзор уровней абстрактного описания систем показывает, что выбор подходящего метода формального описания при изучении той или иной реальной системы является всегда наиболее ответственным и трудным шагом в теоретико-системных построениях. Эта часть исследования почти не поддаётся формализации и во многом зависит от эрудиции исследователя, его профессиональной принадлежности, целей исследования и т.д. Наибольшее значение в настоящее время в абстрактной теории систем придаётся теоретико-множественному, абстрактно-алгебраическому и динамическому уровням описания систем.

2.2. КАНОНИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Стандартное представление любой системы связано, прежде всего, с заданием (описанием) множества показателей (характеристик, свойств и т.п.), на основе которых впоследствии оценивается её качество или эффективность функционирования в смысле достижения цели. Такие показатели называют критериальными, поскольку если задать требования (предпочтительность) для их значений (максимальное и минимальное значение), то показатели становятся критериями, которые в свою очередь используются с целью оптимизации системы.

Первоначально задача количественного оценивания систем формировалась в терминах критерия оптимальности K в виде:

$$K \rightarrow \max y_i, \quad i = \overline{1, n},$$

где y_i – критериальные показатели системы; n – количество показателей системы.

Однако, большинство частных показателей y_i связаны между собой так, что повышение качества системы по одному показателю ведёт к понижению качества системы по другому показателю. Такая постановка задачи оценивания была признана некорректной для большинства приложений.

Таким образом, наличие неоднородных связей между отдельными показателями качества сложных систем приводит к проблеме корректности критерия оптимизации, т.е. к необходимости идти на компромисс и выбирать для каждой характеристики не оптимальное значение, а меньшее, но такое, при котором другие показатели тоже имели бы приемлемые значения. Поэтому критерии разделяют на три специальных класса, которые будут рассмотрены ниже.

Поскольку критерии определяют существенные свойства систем, то эти критерии в соответствии со свойствами можно разделить на 3 группы:

1. Общесистемные свойства и критерии – целостность, устойчивость, наблюдаемость, управляемость, открытость, динамичность.

2. Структурные свойства и критерии – состав, организация, сложность, масштабность, централизованность.

Критерии, относящиеся к пунктам 1 и 2, носят общее название «критерии качества систем».

3. Функциональные (поведенческие) свойства и критерии функционирования – результативность (Э), ресурсоёмкость (R), оперативность (O).

При таком рассмотрении первая и вторая группа свойств определяют критерии качества системы, а свойства, которые характеризуют процесс функционирования системы, можно назвать операционными свойствами или свойствами операций.

Искусственные системы создаются для реализации одной или нескольких операций. Причём требуемый и реально достижимый результаты могут отличаться. Это зависит от условия протекания операции, качества системы и способов достижения требуемого результата. Поэтому при оценке систем принято различать качество системы и эффективность функционирования, т.е. реализации системных процессов.

2.2.1. Критерии качества информационной системы

Каждое i -е качество j -й системы, где $i = \overline{1, n}$, $j = \overline{1, m}$, может быть описано некоторой выходной переменной y_i^j . Она отражает определённое качество (свойство) системы, и её значение характеризует меру этого качества (свойства). Эту меру называют показателем свойства или частным критериальным показателем качества системы.

Показатель y_i^j может принимать значения $y_i^j \in \{y^{\text{доп}}\}$.

Тогда обобщённым показателем качества j -й системы будет некоторый вектор $\overline{Y^j} = \langle Y_1^j, \dots, Y_n^j \rangle$.

Причём обобщённый показатель – это именно вектор, а не множество, так как между отдельными свойствами могут существовать определённые связи, которые сложно описать в теории множеств.

Требуемое качество системы задаётся некоторыми правилами (условиями), которым должны удовлетворяться показатели существенных свойств (частные критерии качества), а проверка выполнения правил или условий называется оцениванием качества системы. Критерий качества – это показатель существенных свойств системы и правило его оценивания.

Предположим, что существует некоторая гипотетическая система, которая является идеальной.

Введём понятие области адекватности, как некоторой области показателей существенных свойств. Геометрическая интерпретация описания (оценки) качества или эффективности функционирования системы для двух критериальных показателей приведена на рис. 2.1.

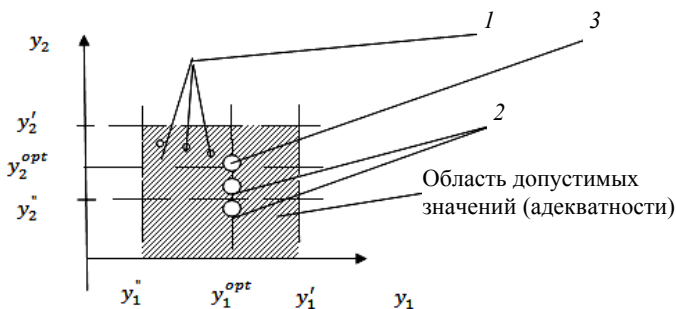


Рис. 2.1. Геометрическая интерпретация критериев качества систем:

1 – критерий пригодности; 2 – критерий оптимальности;

3 – критерий превосходства

Тогда в общем виде область адекватности определяется (σ – радиус области адекватности) как модуль нормируемой разности между допустимыми $y_i^{\text{доп}} = |y_i' - y_i''|$ показателями качества.

Все критерии качества в общем случае можно разделить на 3 класса:

1. Критерий пригодности ($K^{\text{приг}}$).

Это правило, согласно которому некоторая j -я система считается пригодной, если значения всех i -х частных показателей Y_i^j этой системы принадлежат области адекватности, а радиус области адекватности соответствует допустимым значениям всех частных показателей.

$$K^{\text{приг}} : (\forall_i) (y_i^j \in \sigma | \sigma_i \rightarrow y_i^{\text{доп}}, i = 1, n).$$

2. Критерий оптимальности ($K^{\text{опт}}$).

Правило, согласно которому j -я система считается оптимальной по i -му показателю качества, если существует хотя бы один частный показатель качества y_i^j , значение которого принадлежит значению области адекватности σ , а радиус области адекватности по этому показателю оптимален.

$$K^{\text{опт}} : (\exists_i) (y_i^j \in \sigma | \sigma_i = \sigma^{\text{опт}}).$$

3. Критерий превосходства ($K^{\text{прев}}$).

Правило, согласно которому некоторая j -я система считается превосходной, если все значения частных показателей принадлежат области адекватности, а радиус области адекватности оптимален по всем показателям.

$$K^{\text{прев}} : (\forall_i) (y_i^j \in \sigma | \sigma_i \rightarrow \sigma^{\text{опт}}).$$

Из вышесказанного следует, что $K^{\text{прев}} \subset K^{\text{опт}} \subset K^{\text{приг}}$.

2.2.2. Критерии эффективности функционирования информационной системы

Информационные системы создаются именно для выполнения конкретных операций. В общем случае оценка операционных свойств системы проводится в двух аспектах:

- 1) для оценки исхода (результата) операции;
- 2) для оценки алгоритма получения результатов.

Качество исхода операции и алгоритма, обеспечивающее получение этого исхода, оценивается по показателям качества операции, к

которым относятся результативность (Э), ресурсоёмкость (R), оперативность (O).

Рассмотрим эти критерии подробнее.

Результативность (Э) обуславливается полученным целевым эффектом, ради которого функционирует система.

Ресурсоёмкость (R) характеризуется ресурсами всех видов (людскими, техническими, энергетическими и т.д.), используемыми для получения целевого эффекта.

Оперативность (O) определяется расходом времени, потребного для достижения цели операции.

Для количественной оценки исхода операции вводится понятие показателя исхода операции (ПИО) – вектор, отражающий оценки свойств системы:

$$Y_{\text{исх}} = Y_{\text{Э}}, Y_{\text{R}}, Y_{\text{O}}.$$

В совокупности результативность (Э), ресурсоёмкость (R) и оперативность (O) порождают комплексное свойство системы, которое называют эффективностью процесса – $Y_{\text{эф}}$.

$Y_{\text{эф}}$ – степень приспособленности процесса функционирования системы к достижению цели (критерий эффективности функционирования), причём это свойство присуще только операциям, проявляется при функционировании системы и зависит от свойств самой системы и от внешней среды. Выбор критерия эффективности системы – самый ответственный момент в процедуре исследования системы.

Процесс выбора критерия эффективности, как и процесс определения цели является субъективным, творческим, требующим индивидуального подхода. Математическое выражение $Y_{\text{эф}}$ является целевой функцией, т.е. её экстремизация есть отображение цели операции.

Показатели исходов операции, на основе которых формируется критерий эффективности – показатели эффективности. В отдельных операциях показатель исхода этой операции может прямо выступать в качестве критерия эффективности.

На практике существует 3 группы показателей эффективности в зависимости от условий формализации системы:

1. Условия определенности: если ПИО отражает один строго определённый исход детерминированной операции.

2. Условия риска определяются вероятностью или стохастическим характером исходной операции. Показатели – дискретными (непрерывными), случайными величинами с известным законом распределения в вероятностной операции.

3. Критерий условий неопределённости: ПИО являются случайными величинами, законы распределения которых неизвестны.

Уточним описание критериев качества через критерии (показатели) функционирования (эффективности):

1. Критерии пригодности для оценки детерминированных систем с детерминированными операциями $K^{\text{приг}} : (\forall i)(y_i^j \in \sigma_i | \sigma_i \rightarrow y_i^{\text{доп}}, i \in \mathcal{E}, R, O)$.

Этот критерий пригодности определяет правило, по которому операция считается эффективной, если все части системы принадлежат области адекватности.

2. Критерии оптимальности $K^{\text{опт}} : (\exists i)(y_i^j \in \sigma_i | \sigma_i \rightarrow y_i^{\text{опт}}, i \in \mathcal{E}, R, O)$.

Эти критерии определяют правило, по которому операция считается эффективной, если все частные показатели принадлежат области адекватности, а радиус – оптимален.

Для оценки вероятностных операций:

1) критерий пригодности имеет вид:

$$K^{\text{приг}} : P_{\text{дц}}(Y_{\text{эф}}) \geq P_{\text{дц}}^{\text{треб}}(Y_{\text{эф}}).$$

Определяет правило, по которому операция считается эффективной, если вероятность достижения цели по показателям эффективности не меньше требуемой вероятности по этим же показателям эффективности.

2) критерий оптимальности имеет вид:

$$K^{\text{опт}} : P_{\text{дц}}(Y_{\text{эф}}) = P_{\text{дц}}^{\text{треб}}(Y_{\text{опт}}).$$

Операция эффективная, если вероятность достижения цели равна вероятности достижения цели по оптимальным значениям этих показателей.

Основной проблемой оценки эффективности вероятностных операций является неясность способа определения требуемых вероятностей как следствие отсутствие достаточной статистики. Применение методов классификации теории вероятности допустимо при повторяемости опытов и одинаковости условий проведения этих опытов, а эти требования в сложных системах выполняются не всегда, для вероятностных операций проблематично априорное знание законов распределения исходов.

Наибольшие трудности представляет оценка эффективности систем в условиях неопределённости. Для решения этой задачи разработано несколько специальных подходов. Порядок оценки эффективности систем в условиях неопределённости операций составляет один из разделов в теории принятия решений.

Для решения проблемы корректности критериев были разработаны методы количественной оценки систем, а именно:

– методы теории полезности (основаны на аксиоматическом использовании отношения предпочтения, на множестве векторных оценок системы);

– методы векторной оптимизации (основаны на эвристическом понятии векторного критерия качества систем);

– методы ситуационного управления (инженерии знаний, которые основаны на построении семиотических моделей оценки систем). В таких моделях система предпочтений формируется в виде набора логических правил, по которым может быть осуществлён выбор альтернатив. Понятие векторного критерия в явном виде не используется.

Эти три группы методов основаны на трёх важных особенностях оценивания систем:

1. Считается, что не существует системы наилучшей в независимом от ЛПР смысле (субъективность);

2. Считается, что не существует оптимальной системы для всех целей и воздействий внешней среды;

3. Методы исследования операций не удовлетворяют требованиям, предъявляемым к задачам оценивания сложных организационных систем.

2.3. ТЕОРЕТИКО-МНОЖЕСТВЕННЫЕ МОДЕЛИ ИНФОРМАЦИОННЫХ СИСТЕМ

Желая получить математическую модель процесса функционирования системы, охватывающую широкий класс реальных объектов, в общей теории систем исходят из общих предположений о характере функционирования системы:

1) система функционирует во времени; в каждый момент времени система может находиться в одном из возможных состояний;

2) на вход системы могут поступать входные сигналы;

3) система способна выдавать выходные сигналы;

4) состояние системы в данный момент времени определяется предыдущими состояниями и входными сигналами, поступившими в данный момент времени и ранее;

5) выходной сигнал в данный момент времени определяется состояниями системы и входными сигналами, относящимися к данному и предшествующим моментам времени.

Первое из перечисленных предположений отражает динамический характер процесса функционирования в пространстве и времени. При этом процесс функционирования протекает как последовательная смена состояний системы под действием внешних и внутренних причин.

Второе и третье предположения отражают взаимодействие системы с внешней средой.

В четвёртом и пятом предположениях отражается реакция системы на внутренние факторы и воздействия внешней среды: последствие и принцип физической реализуемости.

Многим явлениям и процессам свойственно последствие, вследствие которого тенденции, определяющие поведение системы в будущем, зависят не только от того, в каком состоянии находится система в настоящий момент времени, но в той или иной степени от её поведения в предыдущие моменты времени (например, усвоение студентом сложных дисциплин – теории систем, теории построения АСУ, исследования операций, теории массового обслуживания и др. – зависит от степени усвоения курса теории вероятностей и математической статистики, а ещё дальше – от знания курса высшей математики).

Принцип физической реализуемости заключается в следующем: система не реагирует в данный момент времени на «будущие» факторы и воздействия внешней среды.

2.3.1. Система как отношение на абстрактных множествах

Одним из центральных понятий теории систем является понятие системы, определённое в теоретико-множественных терминах:

$$S \subset \otimes \{V_i, i \in I\},$$

где V_i – вес компоненты; $i \in I$ – индекс компонент декартова произведения $\otimes V_i$, называемых объектами системы S ; I – множество индексов.

В кибернетике наибольший интерес представляют системы с двумя объектами – входным объектом X и выходным объектом Y : $S \subset X \otimes Y$.

Основными причинами определения системы как теоретико-множественного отношения являются следующие:

1. Система определяется в терминах её наблюдаемых свойств или, точнее говоря, в терминах взаимосвязей между этими свойствами, а не тем, что они на самом деле собой представляют (т.е. не с помощью физических, химических, биологических, социальных или других явлений). Это вполне согласуется с природой системных исследований, направленных на выяснение организации и взаимосвязи элементов системы, а не на изучение конкретных механизмов в системе.

2. Определение системы является предельно общим. Конечно, различным системам отвечают и различные способы задания описания (дифференциальные уравнения, булева алгебра, графы и т.д.). В условиях предельно нечёткой информации, когда систему удаётся описать

лишь качественно, все словесные утверждения в силу их лингвистических функций определяют общую математическую модель системы. Действительно, каждое высказывание содержит две основные лингвистические категории: термы (денотаты) и функторы. Напомним, что термы используются для обозначения объектов, а функторы – для обозначения отношения между ними. И для каждого правильного множества словесных утверждений существует отношение (в математическом смысле слова), описывающее формальную взаимосвязь между объектами. Таким образом, система всегда является отношением в общем виде, а уже более узкие классы систем определяются более точно своими специфическими средствами.

3. Системы часто задаются с помощью некоторых уравнений относительно соответствующих переменных. Каждой такой переменной можно поставить в соответствие некоторый объект системы, описывающий область значений соответствующей переменной. Утверждая, что система описывается системой уравнений относительно некоторого множества переменных, в сущности считают, что система есть отношение над соответствующими объектами, порождёнными этими переменными (по одному объекту на каждую переменную, область значений которой он представляет). При этом любая комбинация элементов этих объектов, принадлежащая этому отношению, удовлетворяет исходной системе уравнений.

Под отношением понимается подмножество конечной декартовой степени $A^n = A \times A \times \dots \times A$ данного множества A , т.е. подмножество систем (a_1, a_2, \dots, a_n) из n элементов множества A .

Подмножество $R \subset A^n$ называется n -местным или n -арным отношением в множестве A . Число n называется рангом или типом отношения R . Множество всех n -арных отношений в множестве A относительно операций \cup и \cap является булевой алгеброй.

Для построения теории систем на теоретико-множественном уровне необходимо наделять систему как отношение некоторой дополнительной структурой. Это можно сделать двумя способами:

– ввести дополнительную структуру для элементов объектов системы; например, рассматривать сам элемент $v_i \in V_i$ как некоторое множество с подходящей структурой;

– ввести структуру непосредственно для самих объектов системы $V_i, i \in I$.

Первый способ приводит к понятию (абстрактных) временных систем, а второй – к понятию алгебраических систем.

2.3.2. Временные, алгебраические и функциональные системы

Временные системы. Если элементы одного из объектов системы есть функции, например $v: T_v \rightarrow A_v$, то этот объект называют функциональным. В случае, когда области определения всех функций для данного объекта V одинаковы, т.е. каждая функция $v \in V$ является отображением T в A , $v: T \rightarrow A$, то T называется индексирующим множеством для v , а A – алфавитом объекта T . Если индексирующее множество линейно упорядочено, то его называют множеством моментов времени. Функции, определённые на множествах моментов времени, принято называть (абстрактными) функциями времени. Объект, элементами которого являются временные функции, называют временным объектом, а системы, определённые на временных объектах, – временными системами.

Особый интерес для исследования представляют системы, у которых элементы и входного и выходного объектов определены на одном и том же множестве: $X \subset A^T$ и $Y \subset B^T$. В этом случае под системой понимается отношение $S \subset A^T \times B^T$.

Алгебраические системы. Другой путь наделения объектов системы математическими структурами состоит в определении одной или нескольких операций, относительно которых V становится алгеброй. В самом простейшем случае определяется бинарная операция $R: V \times V \rightarrow V$ и предполагается, что в V можно выделить такое подмножество W , зачастую конечное, что любой элемент $v \in V$ можно получить в результате применения операции R к элементам из W или к элементам, уже построенным из элементов множества подобным образом. В этом случае W называют множеством производящих элементов или алфавитом объекта, а его элементы – символами, а элементы объекта V – словами. Если R есть операция сочленения, то слова – это просто последовательности элементов алфавита W .

Необходимо иметь в виду, что алфавит временного объекта – это не совсем то же самое, что алфавит алгебраического объекта. Для объектов с конечными алфавитами – это обычно одни и те же множества. Но как только алфавит становится бесконечным, возникают трудности: множество производящих элементов и область функций времени оказываются различными множествами, в общем случае даже разной мощности.

Итак, системой называется отношение на непустых (абстрактных) множествах:

$$S \subset \times \{V_i, i \in I\}.$$

Если множество индексов конечно, то выражение можно переписать в виде

$$S \subset V_1 \times V_2 \times \dots \times V_n.$$

Пусть $I_x \subset I$ и $I_y \subset I$ образуют разбиение множества I , т.е. пусть $I_x \cap I_y = \emptyset$ и $I_x \cup I_y = I$.

Множество $X = \otimes \{V_{i_x}, I \in I_x\}$ называется входным объектом, а множество $Y = \otimes \{V_{i_y}, i \in I_y\}$ – выходным объектом системы. Тогда система S определяется отношением $S \subset X \times Y$ и называется системой «вход – выход» («чёрный ящик»).

Если S является функцией $S: X \rightarrow Y$, то система называется функциональной.

Временные системы в терминах «ВХОД – ВЫХОД». Первая часть первого предположения о характере функционирования систем гласит: система функционирует во времени. Множество моментов времени t , в которые рассматривается функционирование системы, обозначим T , $t \in T$. Множество T будем считать подмножеством множества действительных чисел. В частности, оно может быть конечным или счётным. В зависимости от характера множества T различают: дискретное, непрерывное и дискретно-непрерывное время. На практике часто представляют интерес только такие множества T , элементы которых располагаются в изолированных точках числовой оси. В этом случае говорят, что система функционирует в дискретном времени, например контактные схемы, конечные автоматы, вычислительные устройства ЭВМ и т.д. Вместо моментов времени t_0, t_1, \dots часто пишут ряд натуральных чисел $0, 1, 2, \dots$, которые называются тактами.

Множество T представляет собой множество некоторого (конечного или бесконечного) интервала числовой оси. В этом случае говорят, что система функционирует в непрерывном времени, например механические и электрические системы, системы, рассматриваемые в теории автоматического регулирования, и т.д.

Не исключены случаи, когда множество T имеет дискретно-непрерывный характер: на одних интервалах числовой прямой моменты $t \in T$ заполняют их целиком, а на других – располагаются в изолированных точках. Например: 1) метеорологическая ракета при нахождении в состоянии готовности функционирует в непрерывном времени, а при запуске (при работе автомата пуска) можно условно считать, что работает в дискретном времени (реле времени работает дискретно в смысле выдачи команд исполнительным органом по тактам); 2) процесс производства автомобилей на конвейере; конвейер движется не-

прерывно, а готовые автомобили сходят с него в дискретные моменты времени.

Входные сигналы системы. Второе и третье предположения о характере функционирования систем направлены на описание взаимодействия системы с внешней средой. На вход системы могут поступать входные сигналы $x \in X$, где X – множество входных сигналов системы. Входной сигнал, поступивший в момент времени $t \in T$, обозначается $x(t)$.

Возвратимся к примеру с выпуском предприятием однотипных изделий (часто их называют однопродуктовым производством). В такой системе готовность i -го изделия в момент t (автомобиля, часов, велосипеда, телевизора и т.д.) можно описать как поступление очередного сигнала $x(t_i) = 1$. Здесь множество X состоит из одного элемента $x = 1$. Если принять за $X = 0$ сигнал, когда очередное изделие не готово, а за $X = 1$, когда оно готово, то можно считать, что $X = \{0, 1\}$, и в систему входной сигнал поступает в каждый момент $t \in T$. В случае, когда в моменты t_1 оказываются готовыми одновременно несколько изделий (на заводе несколько конвейерных линий), например $0 \leq x \leq x_{\max}$, то множество X – совокупность целых чисел $X = \{0, 1, \dots, X_{\max}\}$.

Входные сигналы могут описываться некоторым набором характеристик. Например, если входными сигналами АСУ аэродромом считать самолеты, поступившие в зону аэродрома, то каждый из них может быть описан: 1) координатами точки взлёта (I, a, ε) (I – наклонная дальность, a – азимут и ε – угол места); 2) вектором скорости (V) ; 3) признаками, характеризующими тип самолета (C) , массу груза (G) , требованиями к аэродромному обслуживанию (δ) и т.д.

В общем случае будем предполагать, что входной сигнал $X_1 \in X_i$, где X – заданные множества ($i = 1, \dots, n$).

Прямое произведение $X = X_1 \times X_2 \times \dots \times X_n$ называется пространством входных сигналов. X_i – элементарные оси. Входной сигнал x представляет собой точку пространства X , описываемую координатами x_1, x_2, \dots, x_n . В общем случае $X_i \subset X$.

При исследовании сложных систем приходится оперировать с группами входных сигналов, поступающих в моменты времени $t_1 < t_2 < \dots < t_k$. Будем предполагать, что множеству X принадлежит и пустой сигнал x_{\emptyset} , означающий отсутствие сигнала в момент t_0 , $x(t_0) = x_{\emptyset}$.

Рассмотрим отображение $x = L(t)$, сопоставляющее каждому $t \in T$ некоторый сигнал $x \in X$ (отображение $f: T \rightarrow X$). Обозначим через T^L

множество моментов времени $T^L \subset T$, такое, что для любого $t' \in T^L$ справедливо $L(t') \neq x_{\emptyset}$. Отображение $x = L(t)$ будем называть входным процессом систем, а совокупность упорядоченных пар (t', x) для всех $t' \in T^L$ (где $x = L(t')$) – входным сообщением.

Чтобы задать конкретный входной процесс $x = L(t)$, достаточно указать соответствующее ему входное сообщение $(t, x_L)_T$.

Интервал времени $t_1 < t < t_2$ будем обозначать (t_1, t_2) , а полуинтервалы $t_1 < t \leq t_2$ и $t_1 \leq t < t_2$ – через $(t_1, t_2]$ и $[t_1, t_2)$, соответственно $t_1 \leq t \leq t_2$ – через $[t_1, t_2]$.

Введём понятие «сужение отображения». Пусть множество X имеет область определения отображения $y = f(x)$. Отображение $y = g(x)$ с областью определения X^* является сужением отображения $f(x)$ на множество X^* в том и только в том случае, когда $X^* \subset X$ и $g(x) = f(x)$ для каждого $x \in X^*$.

Сужение отображения $x = L(t)$ на множество $T \cap (t_1, t_2]$ будем называть фрагментом входного процесса, соответствующим полуинтервалу $(t_1, t_2]$, а совокупность упорядоченных пар (t', x) для всех $t' \in T^L \cap (t_1, t_2]$, где $x = L(t')$ – отрывком входного сообщения, поступающим в систему за полуинтервал $(t_1, t_2]$ и обозначать $(t_1, x_L]_{t_1}^{t_2}$.

Для конечного множества $T^L \cap (t_1, t_2]$, например t_1, t_2, \dots, t_k , входное сообщение имеет вид $(t_1, x_1; t_2, x_2; \dots; t_k, x_k)$.

Множество всевозможных входных сообщений обозначим $\{(t, x_L)_T\}$. Оно определяется множеством входных процессов вида $x = L(t)$, допускаемых условиями функционирования системы. К множеству $\{(t, x_L)_T\}$ будем причислять и пустое входное сообщение $(t, x_L)_T = \emptyset$, для которого $T^L = \emptyset$.

Кроме того, множество $\{(t, x_L)_T\}$ должно удовлетворять ещё одному требованию, связанному с сочленением входных сообщений. Пусть $(t, x_{L1})_T$ и $(t, x_{L2})_T$ – сообщения из множества $\{(t, x_L)_T\}$. Пусть, далее, $t_1 < t_2 < t_3$; $t_1, t_2, t_3 \in T$. Образует отрывки сообщений $(t, x_{L1}]_{t_1}^{t_2}$ и $(t, x_{L2}]_{t_2}^{t_3}$. Совокупность упорядоченных пар (t^*, x^*) можно рассматривать как отрывок $(t, x_L]_{t_1}^{t_3}$ некоторого сообщения $(t, x_L)_T$, образовавшийся в результате сочленения отрывков $(t, x_{L1}]_{t_1}^{t_2}$ и $(t, x_{L2}]_{t_2}^{t_3}$. Сочленение любого числа отрывков входных сообщений из множества

$\{(t, X_i)_T\}$ представляет собой отрывок некоторого входного сообщения, принадлежащего этому множеству.

Выходные сигналы системы. Система способна выдавать выходные сигналы $y \in Y$, где Y – множество выходных сигналов системы. Выходной сигнал, выдаваемый системой в момент времени $t \in T$, обозначается $y(i)$.

$$t^* \in \left(\left\{ T^{L_1} \cap (t_1, t_2) \right\} \cup \left\{ T^{L_2} \cap (t_2, t_3) \right\} \right);$$

$$x^* = \begin{cases} L_1(t^*) & \text{для } t^* \in \left\{ T^{L_1} \cap (t_1, t_2) \right\}; \\ L_2(t^*) & \text{для } t^* \in \left\{ T^{L_2} \cap (t_2, t_3) \right\}. \end{cases}$$

Если выходной сигнал y описывается набором характеристик y_1, y_2, \dots, y_m , таких, что $y \in Y_j, j = 1, \dots, m, Y_j$ – заданные множества, то прямое произведение $Y = Y_1 \times Y_2 \times \dots \times Y_m$ называется пространством выходных сигналов системы. По аналогии с входным процессом введём понятие выходного процесса $y = N(t)$, а также определим выходное сообщение $(t, y_N)_T$ и его отрывок $(t, y_N)_{t_1}^{t_2}$ на полуинтервале $(t_1, t_2]$.

На этом можно считать исчерпанной формальную интерпретацию второго и третьего предположений о характере функционирования систем.

Глобальное состояние и глобальная реакция системы. Пусть для системы S множество её состояний Z , а функция $R: (X \times Z) \rightarrow Y$ такова, что $(x, y) \in S \Rightarrow (\exists_z)[R(x, z) = y]$.

Тогда Z называют множеством или объектом глобальных состояний системы, а элементы множества $z \in Z$ – глобальными состояниями системы. Функция R называется глобальной реакцией системы S . При этом ни на Z , ни на R не накладывается никаких дополнительных условий. В случаях, когда глобальную реакцию системы нельзя определить на всём произведении $X \times Z$, то R оказывается частичной функцией. Таким образом, R можно называть глобальной реакцией системы только тогда, когда она не является частичной функцией. В противном случае её называют частичной глобальной реакцией.

Абстрактные линейные системы. Хотя многие понятия теории систем можно определить, опираясь исключительно на общее понятие системы, получение содержательных математических результатов становится возможным только после введения дополнительных структур. Таким дополнительным понятием является понятие линейности систем.

В соответствии с линейной алгеброй, операции «+» и умножения на скаляр определяются на $X \times Y$ естественным образом:

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2);$$

$$a(x, y) = (ax, ay) \subset X \times Y, a \in A.$$

В теории линейных систем фундаментальную роль играет следующая теорема.

Пусть X и Y – линейные алгебры над одним и тем же полем A . Система $S \subset X \times Y$ является линейной в том случае, когда найдётся такая глобальная реакция $R : X \times Z \rightarrow Y$, что

1. Z есть линейная алгебра над A ;
2. Существует пара таких линейных отображений $R_1 : Z \rightarrow Y$ и $R_2 : X \rightarrow Y$, что для всех $(x, y) \in X \times Y$, $R(x, z) = R_1(x) + R_2(z)$.

Отображение R называют линейной глобальной реакцией системы тогда, и только тогда, когда

1. R согласуется с S , т.е.

$$(x, y) \in S \Rightarrow (\exists z)[R(x, z) = y].$$

2. Z является линейной алгеброй над полем A скаляров линейных алгебр X и Y .

Существуют два таких линейных отображений $R_1 : Z \rightarrow Y$ и $R_2 : X \rightarrow Y$, что для любых $(x, y) \in X \times Y$, $R(x, z) = R_1(x) + R_2(z)$

В этом случае Z называют линейным объектом глобальных состояний системы, отображение $R_1 : Z \rightarrow Y$ – глобальной реакцией на состояние, а $R_2 : X \rightarrow Y$ – глобальной реакцией на вход.

2.4. МОДЕЛИРОВАНИЕ СИСТЕМ СЕТЯМИ ПЕТРИ

Модель – это представление, как правило, в математических терминах наиболее характерных черт изучаемого объекта или системы. Одним из самых распространённых инструментов для математического моделирования и исследования информационных процессов и систем являются *сети Петри*. Цель представления системы в виде сети Петри и последующего анализа этой сети состоит в получении важной информации о структуре и динамическом поведении моделируемой системы. Эта информация может использоваться для оценки моделируемой системы и выработки предложений по её усовершенствованию. Впервые сети Петри предложил немецкий математик Карл Адам Петри.

Сети Петри предназначены для моделирования систем, которые состоят из множества взаимодействующих друг с другом *компонент*. При этом компонента сама может быть системой. Действиям различным компонент системы присущ параллелизм. Примерами таких систем могут служить вычислительные системы, в том числе и параллельные, компьютерные сети, программные системы, обеспечивающие

их функционирование, а также экономические системы, системы управления дорожным движением, химические системы и т.д.

В одном из подходов к проектированию и анализу систем сети Петри используются как вспомогательный инструмент анализа. Здесь для построения системы используются общепринятые методы проектирования. Затем построенная система моделируется сетью Петри, и модель анализируется. Если в ходе анализа в проекте найдены изъяны, то с целью их устранения проект модифицируется. Модифицированный проект затем снова моделируется и анализируется. Этот цикл повторяется до тех пор, пока проводимый анализ не приведёт к успеху.

Другой подход предполагает построение проекта сразу в виде сети Петри. Методы анализа применяются только для создания проекта, не содержащего ошибок. Затем сеть Петри преобразуется в реальную рабочую систему.

В первом случае необходима разработка методов моделирования систем сетями Петри, а во втором случае должны быть разработаны методы реализации сетей Петри системами.

Пусть *мультимножество* – это множество, допускающее вхождение нескольких экземпляров одного и того же элемента.

Определение 2.1. *Сеть Петри* N является четвёркой $N = (P, T, I, O)$, где

$P = \{p_1, p_2, \dots, p_n\}$ – конечное множество *позиций* n ;

$T = \{t_1, t_2, \dots, t_m\}$ – конечное множество *переходов* m ;

$I: T \textcircled{R} P^*$ – входная функция, сопоставляющая переходу T мультимножество его входных позиций P^* ;

$O: T \textcircled{R} P^*$ – выходная функция, сопоставляющая переходу мультимножество его выходных позиций.

Позиция $p \hat{I} P$ называется *входом* для перехода $\hat{I} T$, если $p \hat{I}(t)$.

Символ \hat{I} означает «поставить в соответствие».

Позиция $p \hat{O} P$ называется *выходом* для перехода $\hat{I} T$, если $p \hat{O}(t)$.

Структура сети Петри определяется её позициями, переходами, входной и выходной функциями.

Пример 2.1. Сеть Петри

$$N = (P, T, I, O),$$

$$P = \{p_1, p_2, p_3\},$$

$$T = \{t_1, t_2\},$$

$$I(t_1) = \{p_1, p_1, p_2\}, \quad O(t_1) = \{p_3\},$$

$$I(t_2) = \{p_1, p_2, p_2\}, \quad O(t_2) = \{p_3\}.$$

Использование мультимножеств входных и выходных позиций перехода, а не множеств, позволяет позиции быть *кратным входом* и

кратным выходом перехода соответственно. При этом *кратность* определяется числом экземпляров позиции в соответствующем мультимножестве.

Переход t называется *входом* для позиции p , если p является выходом для t . Переход t называется *выходом* для позиции p , если p является входом для t . Существуют альтернативные, эквивалентные определения сетей Петри. В частности, функции I и O могут быть определены таким образом, чтобы сопоставлять позициям входные и выходные мультимножества переходов соответственно.

Наиболее наглядным представлением сети Петри является её графическое представление, которое представляет собой двудольный, ориентированный мультиграф.

Граф сети Петри обладает двумя типами узлов: *кружок* t , представляющий позицию сети Петри, и *планка* $\%_4$, представляющая переход сети Петри. Ориентированные дуги этого графа (стрелки) соединяют переход с его входными и выходными позициями. При этом дуги направлены от входных позиций к переходу и от перехода к выходным позициям. Кратным входным и выходным позициям перехода соответствуют кратные входные и выходные дуги.

Пример 2.2. Граф сети Петри, определённой в примере 2.1.

Определение 2.2. *Граф сети Петри* есть двудольный, ориентированный мультиграф $G = (V, A)$, где $V = \{v_1, v_2, \dots, v_k\}$ – множество вершин, а $A = \{a_1, a_2, \dots, a_r\}$ – мультимножество направленных дуг, и множество V может быть разбито на два непересекающихся подмножества P и T , справедливо либо $v_j \hat{\in} P$ и $v_s \hat{\in} T$, либо $v_j \hat{\in} T$ и $v_s \hat{\in} P$.

Замечание 2.1. Согласно определению графа сети Петри в нём не возможны дуги между двумя позициями и между двумя переходами.

Замечание 2.2. Теоретико-множественное задание сети Петри $N = (P, T, I, O)$ однозначно определяет граф сети Петри $C = (V, A)$.

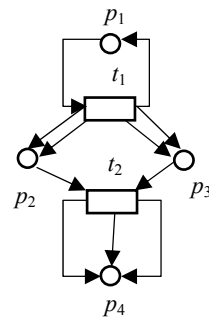
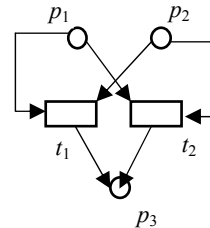
Пример 2.3. Построение графа сети Петри по её теоретико-множественному заданию.

$$N = (P, T, I, O), P = \{p_1, p_2, p_3, p_4\}, T = \{t_1, t_2\},$$

$$I(t_1) = \{p_1\}, O(t_1) = \{p_1, p_2, p_2, p_3, p_3\},$$

$$I(t_2) = \{p_2, p_3\}, O(t_2) = \{p_4, p_4, p_4\}.$$

Рассмотрим маркировку сетей Петри. *Маркировка* – это размещение по позициям сети Петри



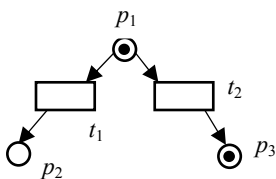
фишек, изображаемых на графе сети Петри точками. Фишки используются для определения выполнения сети Петри. Количество фишек в позиции при выполнении сети Петри может изменяться от 0 до бесконечности.

Определение 2.3. *Маркировка* m сети Петри $N = (P, T, I, O)$ есть функция, отображающая множество позиций P в множество неотрицательных целых чисел Nat (где число из Nat обозначает количество фишек, помещаемых в соответствующую позицию).

Маркировка m может быть также определена как n -вектор $m = \langle m(p_1), m(p_2), \dots, m(p_n) \rangle$, где n – число позиций в сети Петри и для каждого i справедливо $m(p_i) \in Nat$.

Определение 2.4. *Маркированная сеть Петри* $N = (P, T, I, O, m)$ определяется совокупностью структуры сети Петри (P, T, I, O) и маркировки m .

Пример 2.4. Графическое представление маркированной сети Петри



$$m = \langle 1, 0, 1 \rangle.$$

Множество всех маркировок сети Петри бесконечно. Если фишек, помещаемых в позицию слишком много, то удобнее не рисовать фишки в кружке этой позиции, а указывать их количество.

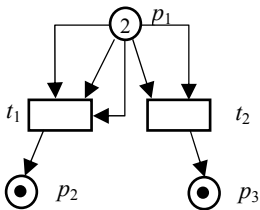
Правила выполнения сетей Петри.

Сеть Петри выполняется посредством запусков переходов. *Запуск* перехода управляется фишками в его входных позициях и сопровождается удалением фишек из этих позиций и добавлением новых фишек в его выходные позиции.

Переход может запускаться только в том случае, когда он разрешён. Переход называется *разрешённым*, если каждая из его входных позиций содержит число фишек, не меньшее, чем число дуг, ведущих из этой позиции в переход (или кратности входной дуги).

Пример 2.5. Разрешённый переход маркированной сети Петри.

В этой сети Петри с маркировкой $m = \langle 2, 1, 1 \rangle$ разрешён переход t_2 и не разрешён переход t_1 .



Пусть функция $\hat{\#}: P \times T \rightarrow Nat$ для произвольных позиции $p \in P$ и перехода $t \in T$ задаёт значение $\hat{\#}(p, t)$, которое совпадает с кратностью дуги, ведущей из p в t , если такая дуга существует, и с нулём в противном случае.

Пусть функция $\hat{\#}: T \times P \rightarrow Nat$ для произвольных перехода $t \in T$ и позиции $p \in P$ задаёт

значение $\hat{\#}(t, p)$, которое совпадает с кратностью дуги, ведущей из t в p , если такая дуга существует, и с нулём, в противном случае.

Пример 2.6. Функции $\hat{\#}$ и $\#$ для сети Петри из примера 4.

$$\begin{aligned} \hat{\#}(p_1, t_1) &= 3, \hat{\#}(p_2, t_1) = 0, \hat{\#}(p_3, t_1) = 0, \\ \hat{\#}(p_1, t_2) &= 2, \hat{\#}(p_2, t_2) = 0, \hat{\#}(p_3, t_2) = 0, \\ \#(t_1, p_1) &= 0, \#(t_1, p_2) = 1, \#(t_1, p_3) = 0, \\ \#(t_2, p_1) &= 0, \#(t_2, p_2) = 0, \#(t_2, p_3) = 1. \end{aligned}$$

Определение 2.5. Переход $t \hat{T}$ в маркированной сети Петри $N = (P, T, I, O, m)$ разрешён, если для всех $p \in I(t)$ справедливо $m(p) = \hat{\#}(p, t)$.

Запуск разрешённого перехода $t \hat{T}$ из своей входной позиции $p \in I(t)$ удаляет $\hat{\#}(p, t)$ фишек, а в свою выходную позицию $p' \in O(t)$ добавляет $\#(t, p')$ фишек.

Пример 2.7. Запуск разрешённого перехода в сети Петри.



Сеть Петри до запуска перехода t_1

Сеть Петри после запуска перехода t_1

Запуск перехода заменяет маркировку m сети Петри на новую маркировку m' .

Определение 2.6. Переход t в маркированной сети Петри с маркировкой m может быть запущен всякий раз, когда он разрешён и в результате запуска разрешённого перехода t образуется новая маркировка m' , определяемая следующим соотношением:

$$m'(p) = m(p) - \hat{\#}(p, t) + \#(t, p) \text{ для всех } p \in P.$$

Пример 2.8. Преобразование маркировки сети Петри в примере 6.

Переход t_1 преобразует маркировку $m = \langle 5, 1 \rangle$ в маркировку $m' = \langle 2, 3 \rangle$.

Запуски могут осуществляться до тех пор, пока существует хотя бы один разрешённый переход. Когда не останется ни одного разрешённого перехода, выполнение *прекращается*.

Если запуск произвольного перехода t преобразует маркировку m сети Петри в новую маркировку m' , то будем говорить, что m' достижима из m посредством запуска перехода t . Это понятие очевидно образом обобщается для случая последовательности запус-

ков разрешённых переходов. Через $R(N, m)$ обозначим множество всех достижимых маркировок из начальной маркировки m в сети Петри N .

Рассмотрим ряд примеров моделирования систем сетями Петри, позволяющих дать представление о большом классе систем, которые можно моделировать сетями Петри, об используемом методе моделирования и о свойствах, которыми должны обладать моделируемые системы. Особое внимание будет уделено системам из области аппаратного и программного обеспечения.

Представление системы сетью Петри основано на двух основополагающих понятиях: *событиях* и *условиях*. Возникновением событий управляет состояние системы, которое может быть описано множеством условий. Условие может принимать либо значение «истина», либо значение «ложь».

Возникновение события в системе возможно, если выполняются определённые условия – *предусловия* события. Возникновение события может привести к выполнению других условий – *постусловий* события. В качестве примера рассмотрим следующую ниже задачу моделирования.

Пример 2.9. Моделирование системы автомат-продавец.

Автомат-продавец находится в состоянии ожидания до тех пор, пока не появится заказ, который он выполняет и посылает на доставку. Условиями для такой системы являются:

- а) автомат-продавец ждёт;
- б) заказ прибыл и ждёт;
- в) автомат-продавец выполняет заказ;
- г) заказ выполнен.

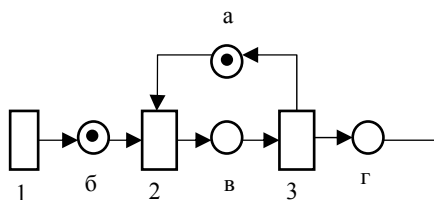
Событиями для этой системы являются:

1. Заказ поступил.
2. Автомат-продавец начинает выполнение заказа.
3. Автомат-продавец заканчивает выполнение заказа.
4. Заказ посылается на доставку.

Для перечисленных событий можно составить следующую таблицу их пред- и постусловий.

Событие	Предусловия	Постусловия
1	нет	б
2	а, б	в
3	в	г, а
4	г	нет

Такое представление системы легко моделировать сетью Петри. В сети Петри условия моделируются позициями, события – переходами. При этом входы перехода являются условиями соответствующего события; выходы – постусловиями. Возникновение события моделируется запуском соответствующего перехода. Выполнение условия представляется фишкой в позиции, соответствующей этому условию. Запуск перехода удаляет фишки, представляющие выполнение условий и образует новые фишки, которые представляют выполнение постусловий.



Пример 2.10. Моделирование последовательной обработки запросов сервером базы данных. Сервер находится в состоянии ожидания до тех пор, пока от пользователя не поступит запрос, который он обрабатывает и отправляет результат такой обработки пользователю.

Условиями для такой системы являются:

- а) сервер ждёт;
- б) запрос поступил и ждёт;
- в) сервер обрабатывает запрос;
- г) запрос обработан.

Событиями для этой системы являются:

1. Запрос поступил.
2. Сервер начинает обработку запроса.
3. Сервер заканчивает обработку запроса.
4. Результат обработки отправляется.

Таблица пред- и постусловий для перечисленных событий совпадает с аналогичной таблицей для системы автомат-продавец из предыдущего примера. Сеть Петри, моделирующая последовательную обработку запросов сервером базы данных, совпадает с аналогичной сетью Петри для системы автомат-продавец.

Пример 2.11. Моделирование параллельной обработки запросов сервером базы данных. Сервер находится в состоянии ожидания до тех пор, пока от пользователя не поступит запрос, который он обрабатывает и отправляет результат пользователю. Особенность состоит в том, что он может обрабатывать одновременно два запроса с помощью двух своих процессорных элементов ПЭ1 и ПЭ2.

Условиями для такой системы являются:

- а1) ПЭ1 ждёт;
- а2) ПЭ2 ждёт;
- б) запрос поступил и ждёт;
- в1) ПЭ1 обрабатывает запрос;
- в2) ПЭ2 обрабатывает запрос;
- г) запрос обработан.

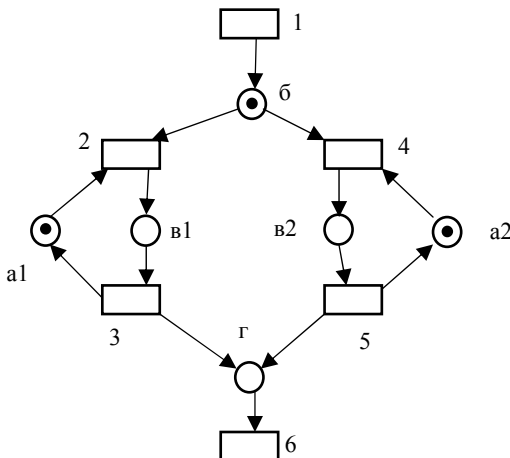
Событиями для этой системы являются:

1. Запрос поступил.
2. ПЭ1 начинает обработку запроса.
3. ПЭ1 заканчивает обработку запроса .
4. ПЭ2 начинает обработку запроса.
5. ПЭ2 заканчивает обработку запроса .
6. Результат обработки отправляется.

Для перечисленных событий можно составить следующую таблицу их пред- и постусловий.

Событие	Предусловия	Постусловия
1	нет	б
2	а1, б	в1
3	в1	г, а1
4	а2, б	в2
5	в2	г, а2
6	г	нет

Сеть Петри, моделирующая эту систему, имеет вид:



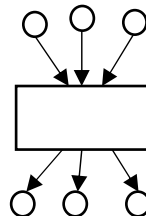
Рассмотрим понятия одновременности и конфликта. Важная особенность сетей Петри – это их *асинхронная* природа. В сети Петри отсутствует измерение времени или течение времени. Это отражает философский подход к понятию времени, утверждающий, что важнейшим свойством времени является частичное упорядочение событий. При этом считается менее важным протяжённость во времени отдельных событий.

Выполнение сети Петри (или поведение моделируемой системы) рассматривается здесь как *последовательность* дискретных событий. Порядок появления событий является одним из возможных. Если в какой-то момент времени разрешено более одного перехода, то любой из них может стать «следующим» запускаемым. Тем самым, выбор запускаемого перехода осуществляется недетерминированным образом, т.е. случайно.

Запуск рассматривается как мгновенное событие. Одновременный запуск нескольких различных переходов невозможен. Моделируемое таким образом событие называется *примитивным*.

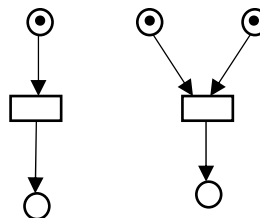
Замечание 2.3. Переходы в сети Петри, моделирующей некоторую систему, представляют её *примитивные* события (длительность которых считается равной 0), и в один момент времени может быть запущен только один разрешённый переход.

Непримитивными называются такие события, длительность которых отлична от нуля. Они могут пересекаться во времени. Непримитивные события не могут моделироваться переходами. Непримитивное событие может быть промоделировано сетью Петри, например, с помощью двух примитивных событий: «начало непримитивного события», «конец непримитивного события» и условия «непримитивное событие происходит». Иногда непримитивные события представляют на графе сети Петри в виде большого прямоугольника.



Такой прямоугольник может иметь большое значение при моделировании сложных систем на нескольких иерархических уровнях, так как он позволяет выделить в отдельные элементы сети целые подсети, тем самым существенно упростить исходную сеть Петри.

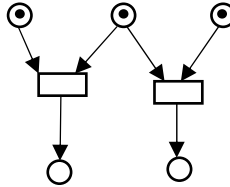
Моделирование параллельного возникновения независимых событий системы в сети Петри демонстрируется на рисунке.



В этой ситуации два перехода являются разрешёнными и не влияют друг на друга в

том смысле, что могут быть запущены один вслед за другим в любом порядке. Такие переходы называются *одновременными* и моделируют параллельное возникновение событий.

Другая ситуация в приведённой ниже сети Петри. В ней одновременность переходов невозможна.



Эти два перехода находятся в *конфликте*, т.е. запуск одного из них удаляет фишку из общей входной позиции и тем самым запрещает запуск другого. Таким образом, моделируются взаимоисключающие события системы.

Замечание 2.4. Области, в которых сети Петри представляются идеальным инструментом для моделирования, характеризуются тем, что в них события происходят асинхронно и независимо.

2.5. МОДЕЛИРОВАНИЕ АППАРАТНОГО ОБЕСПЕЧЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Аппаратное обеспечение вычислительных систем (ВС) можно рассматривать на нескольких уровнях. На низком уровне ВС построены из простых устройств памяти и вентилях; на более высоком уровне в качестве основных компонент системы используются функциональные блоки и регистры. На еще более высоком уровне целые вычислительные системы могут быть компонентами сети ВС. Одним из сильных свойств сетей Петри является их способность моделировать каждый из этих уровней.

Рассмотрим конечные автоматы и их моделирование сетями Петри.

На низком уровне вычислительные системы могут быть описаны *автоматами*. Автомат – это пятёрка (Q, S, D, d, G) , где

Q – конечное множество состояний $\{q_1, q_2, \dots, q_k\}$;

S – конечный входной алфавит;

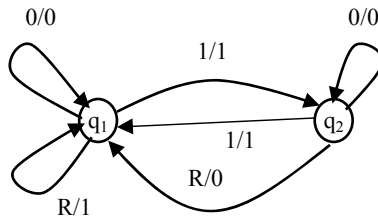
D – конечный выходной алфавит;

$d: Q \times S \rightarrow Q$ – функция следующего состояния, отображающая текущее состояние и текущий вход в следующее состояние;

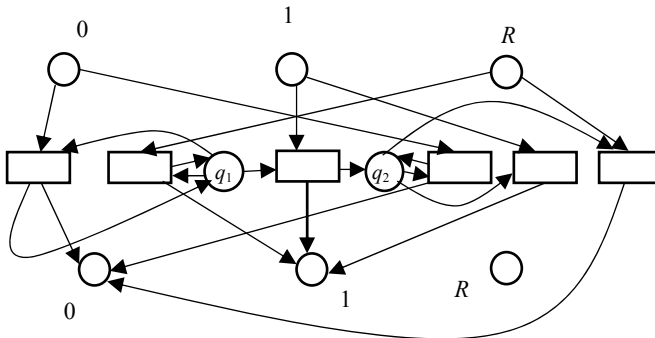
$G: Q \times S \rightarrow D$ – функция выхода, отображающая текущее состояние и текущий вход в выходной символ.

Автоматы часто представляют в виде графов переходов. В графе переходов состояния представляются кружками, являющимися вершинами. Дуга из состояния q_1 в q_2 , помеченная a/b , означает, что, находясь в состоянии q_1 , автомат при входе a перейдёт в состояние q_2 , выдавая при этом символ b . Формально следовало бы записать $d(q_1, a) = q_2$ и $G(q_1, a) = b$.

Пример 2.12. Автомат, вычисляющий чётность количества единиц во входном двоичном числе. Пусть $Q = \{q_1, q_2\}$, $S = D = \{0, 1, R\}$, где R – признак конца числа (символ сброса). Автомат начинает работу в состоянии q_1 . Выход копирует вход до тех пор, пока входным символом не окажется символ сброса R . Выходом для символа сброса будет 0 в случае нечётности и 1 – в случае чётности:



Метод моделирования конечных автоматов. Представим каждый входной и выходной символ, а также каждое состояние автомата позицией в сети Петри. Текущее состояние отмечается фишкой; все остальные позиции пусты. Теперь для определения переходов из состояния в состояние можно ввести переходы сети следующим образом. Для каждой пары (состояние, входной символ) мы определяем переход, входными позициями которого являются позиции, соответствующие состоянию и входному символу, а выходными позициями – позиции, соответствующие следующему состоянию и выходу.



Замечание 2.5. Осуществлённое моделирование конечных автоматов сетями Петри показывает возможность применения сетей Петри для разработки аппаратного обеспечения ВС низкого уровня.

Замечание 2.6. Одно из основных преимуществ моделей автоматов в виде сетей Петри перед самими автоматами состоит в том, что они существенно упрощают композицию автоматов, как последовательную, так и параллельную.

Способность сетей Петри моделировать параллелизм и строить модель системы путём довольно простого объединения моделей подсистем делает их весьма полезным инструментом моделирования сложной аппаратуры вычислительных систем.

2.6. ПРИНЯТИЕ РЕШЕНИЙ КАК ЗАДАЧА СИСТЕМНОГО АНАЛИЗА

Методика системного анализа разрабатывается и применяется, если у лица, принимающего решение (ЛПР), нет необходимых сведений об определённой ситуации, позволяющих её формализовать и найти решение задачи.

В этой ситуации помогает представление объекта в виде системы, привлечение экспертов в различных областях знаний, организация мозговых атак и т.д. Рекомендуется применять различные методы описания систем для создания наиболее эффективного набора методов для данной задачи. Для организации такого процесса определяется часть этапов, выбираются методы для этих этапов, определяются ключевые точки. Часть выделенных и упорядоченных этапов с методами их выполнения представляет собой методику системного анализа.

Под термином *решение* будем понимать совокупность рассматриваемых возможностей, которые тем или иным образом выделены человеком, делающим выбор. Или решением будем называть ответ, полученный в ходе поиска; например, один или несколько выбранных вариантов, а также результат анализа выбранной проблемы или поставленной задачи.

Принятие решений в профессиональном отношении представляет собой особый вид человеческой деятельности, который состоит в обоснованном выборе наилучшего в некотором смысле варианта или нескольких предпочтительных вариантов из имеющихся возможных. По-английски этот термин звучит, как *decision making*, т.е. буквально означает «делание» или создание решения, что более адекватно смыслу этого словосочетания.

Теория принятия решений как самостоятельное научное направление стала складываться в середине XX в. в рамках методологии системного анализа, хотя самые первые работы по исследованию голосования как способа коллективного выбора появились ещё в конце XVII в.

Основное назначение теории принятия решений состоит в разработке методов и средств, позволяющих одному человеку или группе лиц сформулировать множество возможных вариантов решения проблемы, сравнить их между собой, найти среди них лучшие или допустимые варианты, которые удовлетворяют тем или иным требованиям (ограничениям), и при необходимости объяснить сделанный выбор.

Теория принятия решений может оказать существенную помощь в анализе и решении сложных проблем, но лишь тогда, когда её методологические и математические средства применяются «правильно», соответственно их возможностям, не преувеличивая и не умаляя их роли в процессе нахождения решения. Поэтому теорию принятия решений правильнее было бы назвать теорией поиска и обоснованного выбора наиболее предпочтительных для человека вариантов решения проблемы.

Существуют две противоположные точки зрения на роль формальных методов при решении практических проблем выбора. Люди, профессионально не владеющие математическими методами, нередко считают, что любая проблема может быть формально переведена на язык математики и потом решена её средствами. Другие полностью отвергают такие возможности. Действительность же гораздо сложнее этих крайних утверждений.

Любые ситуации, требующие принятия решения, содержат, как правило, большое число неопределённых факторов, которые оказывают влияние, как на формальную постановку задачи, так и на средства её решения. Эти неопределённые факторы можно в самом общем виде разбить на три группы.

Прежде всего, это так называемая *неопределённость природы*, т.е. факторы людям попросту неизвестные или от них не зависящие.

Затем – *неопределённость человека*, который может вести себя непоследовательно, противоречиво, допускать ошибки, зависеть от других лиц (партнёров, противников и т.д.), чьи действия он не может полностью учесть или предвидеть.

И наконец, *неопределённость целей*, которые могут различаться и не совпадать друг с другом. Например, авиаконструкторы, проектируя самолет, должны учитывать его целевое назначение, заданные показатели скорости, грузоподъёмности и дальности полета, условия безопасности и комфортности для экипажа и пассажиров, факторы экономичности и технологичности производства и эксплуатации самолета, экологические требования и многие другие обстоятельства.

Ясно, что полностью свести подобные задачи с неопределённостью к корректно поставленным математическим задачам нельзя в принципе. Чтобы сделать возможным их решение, надо как-то ограничить, уменьшить или, как говорят, «снять» неопределённость. Для это-

го проводится содержательный анализ проблемной ситуации, делаются какие-либо предположения и вводятся упрощения в постановку задачи. И именно средства, входящие в состав тех или иных методов принятия решений, очень часто позволяют получить дополнительную информацию, нужную для формализации реальной проблемной ситуации и приведения её к виду, пригодному для использования математических методов и получения приемлемого результата.

Говоря о практической применимости методов принятия решений, следует особенно подчеркнуть, что должны существовать как объективные внешние обстоятельства, так и субъективные внутренние условия, которые побуждали бы человека – руководителя, ответственного за решение стоящей проблемы, специалиста, аналитика – искать лучшие варианты её решения. Без такой потребности спрос на научно обоснованные методы выбора будет невелик.

Задача принятия решения состоит в формировании множества возможных вариантов, обеспечивающих разрешение проблемной ситуации при существующих ограничениях, и выделении среди этих вариантов одного лучшего или нескольких предпочтительных вариантов, удовлетворяющих предъявляемым к ним требованиям. Формально задачу принятия решения D можно записать в следующем обобщённом виде:

$$D = \{F, A, X, G, P\}.$$

Здесь F – формулировка задачи принятия решения, которая включает в себя содержательное описание стоящей проблемы и при необходимости её модельное представление, определение цели или целей, которые должны быть достигнуты, а также требования к виду окончательного результата.

A – совокупность возможных вариантов (альтернатив), из которых производится выбор. Это могут быть реально существующие варианты, в качестве которых в зависимости от контекста задачи выступают объекты, кандидаты, способы достижения цели, действия, решения и т.п., либо гипотетическое множество всех теоретически возможных вариантов, которое может быть даже бесконечным. Подчеркнём ещё раз, что выбор возникает только тогда, когда имеется не менее двух возможных вариантов решения проблемы.

X – совокупность признаков (атрибутов, параметров) – критерияльных показателей, описывающих варианты и их отличительные особенности. В качестве признаков выступают, во-первых, объективные показатели, которые характеризуют те или иные свойства, присущие вариантам, и которые, как правило, можно измерить; во-вторых, субъективные оценки, которые обычно даются по специально отобранному или сконструированному критерию, отражающим важные для участников выбора черты вариантов. Например, состояние здоровья челове-

ка можно охарактеризовать температурой тела, величиной кровяного давления, отсутствием или наличием боли, её локализацией.

G – совокупность условий, ограничивающих область допустимых вариантов решения задачи. Ограничения могут быть описаны как содержательным образом, так и заданы в виде некоторых формальных требований к вариантам и/или их признакам. Например, это могут быть ограничения на значения какого-либо признака или различная степень характерности (выраженности) признака для тех или иных вариантов, или невозможность одновременного сочетания определённых значений признаков для реально существующих вариантов. Так, если человек здоров, то у него ничего не должно болеть, а температура и кровяное давление должны быть нормальными. Отсутствие ограничений существенно упрощает задачу принятия решения.

P – предпочтения одного или нескольких ЛПР (критерии), которые служат основой для оценки и сравнения возможных вариантов решения проблемы, отбора допустимых вариантов и поиска наилучшего или приемлемого варианта. Достаточно часто для упрощения постановки задачи принятия решения часть информации, описывающей предпочтения ЛПР, превращается в ограничения.

Факторы, характеризующие проблемную ситуацию, условно делятся на две группы: управляемые и неуправляемые. *Управляемые факторы*, выбор которых зависит от ЛПР, – суть поставленные цели, варианты (альтернативы) их достижения, субъективные оценки вариантов и степени достижения целей. *Неуправляемые факторы* не зависят от ЛПР. Они определяют объективные признаки вариантов и отчасти устанавливают ограничения на выбор возможных вариантов.

Факторы подразделяются также на:

- определённые, или детерминированные, с известными и/или заранее заданными точными характеристиками;
- вероятностные, или стохастические, с известными и/или заранее заданными случайными характеристиками;
- неопределённые, или неизвестные, с нечётко определёнными и/или неизвестными характеристиками, но иногда с известной областью изменения их значений.

Модели предпочтения ЛПР. В теории принятия решений используются разные модели формализации предпочтений. Мы рассмотрим 2 модели предпочтений:

1. Реляционная модель предпочтений основана на бинарных отношениях. Бинарные отношения делятся на 3 группы: нейтральные, слабые и сильные.

Нейтральные предпочтения $A_i \approx A_j$ характеризуются симметричным отношением (сходство, эквивалентность; несходство, проти-

воположность) и свидетельствуют либо о некоторой равноценности, либо о неопределённой ценности обоих вариантов для ЛПР.

Слабые предпочтения $A_i \succeq A_j$. Такое отношение задаёт нестрогое превосходство и отражает как различимость, так и одинаковость вариантов для ЛПР. Предполагаются 2 трактовки: A_i не хуже варианта A_j ; A_i по крайней мере такой же как вариант A_j .

Сильные предпочтения $A_i \succ A_j$ – это отношение, выражающее строгое превосходство и явно выраженные различия вариантов: « A_i лучше, чем A_j ».

Таким образом, в рамках реляционной модели предпочтений при необходимости выбора из пары вариантов A_i, A_j в случае нейтральной предпочтительности выбираются оба варианта; в случае нестрогой предпочтительности либо выбирается вариант A_i , либо выбираются или не выбираются оба варианта вместе; в случае строгой предпочтительности выбирается только первый вариант A_i и не выбирается второй вариант A_j .

2. Второй достаточно популярной моделью предпочтения ЛПР является функциональная модель предпочтений. В рамках этой модели предпочтения ЛПР выражаются значением некоторой числовой функции, зависящей от характеристик рассматриваемого варианта. Обозначение $f(\bar{x})$, где f – предпочтения ЛПР, \bar{x} – вектор параметров вариантов.

Такие функции носят разные названия: целевые функции, показатели эффективности, функции ценности и полезности, критерий оптимизации и т.п.

Все многообразие способов выявления предпочтений ЛПР можно свести к трём основным процедурам: *оценке, сравнению и выбору*. Отметим, что сами эти процедуры могут быть как объективного, так и субъективного характера.

Задания для самопроверки

1. Дайте определение математической модели.
2. Перечислите критерии качества системы.
3. Назовите общие предположения о характере функционирования системы в общей теории систем.
4. Опишите подходы, используемые в моделировании систем с помощью сетей Петри.
5. В чём заключается метод моделирования конечных автоматов?
6. Назовите области применения теории принятия решений.

3. МЕТОДЫ И МОДЕЛИ ОПИСАНИЯ (ПРЕДСТАВЛЕНИЯ) СИСТЕМ

3.1. КАЧЕСТВЕННЫЕ МЕТОДЫ ОПИСАНИЯ СИСТЕМ

Качественные методы системного анализа применяются, когда отсутствуют описания закономерностей систем в виде аналитических зависимостей.

Методы типа мозговой атаки. Концепция «мозговой атаки» получила широкое распространение с начала 50-х годов как метод систематической тренировки творческого мышления, нацеленный на открытие новых идей и достижение согласия группы людей на основе интуитивного мышления. Методы этого типа известны также под названиями «мозговой штурм», «конференция идей», а в последнее время наибольшее распространение получил термин «коллективная генерация идей» (КГИ).

Обычно при проведении мозговой атаки или сессий КГИ стараются выполнять определённые правила, суть которых:

- обеспечить как можно большую свободу мышления участников КГИ и высказывания ими новых идей;
- приветствуются любые идеи, если вначале они кажутся сомнительными или абсурдными (обсуждение и оценка идей производится позднее);
- не допускается критика, не объявляется ложной и не прекращается обсуждение ни одной идеи;
- желательно высказывать как можно больше идей, особенно нетривиальных.

Подобием сессий КГИ можно считать разного рода совещания – конструктораты, заседания научных советов по проблемам, заседания специально создаваемых временных комиссий и другие собрания компетентных специалистов.

Методы типа сценариев. Методы подготовки и согласования представлений о проблеме или анализируемом объекте, изложенные в письменном виде, получили название сценария. Первоначально этот метод предполагал подготовку текста, содержащего логическую последовательность событий или возможные варианты решения проблемы, развёрнутые во времени. Однако позднее обязательное требование явно выраженных временных координат было снято, и сценарием стали называть любой документ, содержащий анализ рассматриваемой проблемы или предложения по её решению, по развитию системы не-

зависимо от того, в какой форме он представлен. Как правило, предложения для подготовки подобных документов пишутся вначале индивидуально, а затем формируется согласованный текст.

На практике по типу сценариев разрабатывались прогнозы в некоторых отраслях промышленности. В настоящее время разнообразие сценариев можно считать предложения к комплексным программам развития отраслей народного хозяйства, подготавливаемые организациями или специальными комиссиями.

Сценарий является предварительной информацией, на основе которой проводится дальнейшая работа по прогнозированию развития отрасли или по разработке вариантов проекта. Он может быть подвергнут анализу, чтобы исключить из дальнейшего рассмотрения то, что в учитываемом периоде находится на достаточном уровне развития, если речь идёт о прогнозе, или, напротив, то, что не может быть обеспечено в планируемом периоде, если речь идёт о проекте. Таким образом, сценарий помогает составить представление о проблеме, а затем приступить к более формализованному представлению системы в виде графиков, таблиц для проведения экспертного опроса и других методов системного анализа.

Методы экспертных оценок. Термин «эксперт» происходит от латинского слова, означающего «опытный».

При использовании экспертных оценок обычно предполагается, что мнение группы экспертов надёжнее, чем мнение отдельного эксперта. В некоторых теоретических исследованиях отмечается, что это предположение не является очевидным.

Всё множество проблем, решаемых методами экспертных оценок, делится на два класса. К первому относятся такие, в отношении которых имеется достаточное обеспечение информацией. При этом методы опроса и обработки основываются на использовании принципа «хорошего измерителя», т.е. эксперт – качественный источник информации; групповое мнение экспертов близко к истинному решению. Ко второму классу относятся проблемы, в отношении которых знаний для уверенности в справедливости указанных гипотез недостаточно. В этом случае экспертов уже нельзя рассматривать как «хороших измерителей» и необходимо осторожно подходить к обработке результатов экспертизы во избежание больших ошибок. В литературе в основном рассматриваются вопросы экспертного оценивания для решения задач первого класса.

При обработке материалов коллективной экспертной оценки используются методы теории ранговой корреляции. Для количественной

оценки степени согласованности мнений экспертов применяется коэффициент конкордации

$$W = \frac{12d}{m^2(n^3 - n)},$$

где

$$d = \sum_{i=1}^n d_i^2 = \sum_{i=1}^n \left[\sum_{j=1}^m r_{ij} - 0,5m(n+1) \right]^2,$$

где m – количество экспертов, $j = \overline{1, m}$; n – количество рассматриваемых свойств, $i = \overline{1, n}$; r_{ij} – место, которое заняло i -е свойство в ранжировке j -м экспертом; d_i – отклонение суммы рангов по i -му свойству от среднего арифметического сумм рангов по n свойствам.

Коэффициент конкордации W позволяет оценить, насколько согласованы между собой ряды предпочтительности, построенные каждым экспертом. Его значение находится в пределах $0 \leq W \leq 1$; $W = 0$ означает полную противоположность, а $W = 1$ – полное совпадение ранжировок. Практически достоверность считается хорошей, если $W = 0,7 \dots 0,8$.

Небольшое значение коэффициента конкордации, свидетельствующее о слабой согласованности мнений экспертов, является следствием следующих причин: в рассматриваемой совокупности экспертов действительно отсутствует общность мнений; внутри рассматриваемой совокупности экспертов существуют группы с высокой согласованностью мнений, однако обобщённые мнения таких групп противоположны.

Для наглядности представления о степени согласованности мнений двух любых экспертов A и B служит коэффициент парной ранговой корреляции

$$\rho_{AB} = 1 - \frac{\sum_{i=1}^n \psi_i^2}{\frac{1}{6}(n^3 - n) - \frac{1}{n}(T_A + T_B)},$$

где ψ_i – разность (по модулю) величин рангов оценок i -го свойства, назначенных экспертами A и B : $\psi_i = (R_{A_i} - R_{B_i})$; T_A и T_B – показатели связанных рангов оценок экспертов A и B .

Коэффициент парной ранговой корреляции принимает значения $-1 < \rho < +1$. Значение $\rho = +1$ соответствует полному совпадению

оценок в рангах двух экспертов (полная согласованность мнений двух экспертов), а $\rho = -1$ – двум взаимно противоположным ранжировкам важности свойств (мнение одного эксперта противоположно мнению другого).

Методы типа «Дельфи». Характерный для середины XX в. бурный рост науки и техники вызвал большие перемены в отношении к оценкам будущего развития систем. Одним из результатов этого периода в развитии методов анализа сложных систем явилась разработка методов экспертной оценки, известных в литературе как «методы Дельфи». Название этих методов связано с древнегреческим городом Дельфи, где при храме Аполлона с IX в. до н.э. до IV в. н.э. по преданиям существовал Дельфийский оракул.

Суть метода Дельфи заключается в следующем. В отличие от традиционного подхода к достижению согласованности мнений экспертов путём открытой дискуссии метод Дельфи предполагает полный отказ от коллективных обсуждений. Это делается для того, чтобы уменьшить влияние таких психологических факторов, как присоединение к мнению наиболее авторитетного специалиста, нежелание отказаться от публично выраженного мнения, следование за мнением большинства. В методе Дельфи прямые дебаты заменены тщательно разработанной программой последовательных индивидуальных опросов, проводимых обычно в форме анкетирования. Ответы экспертов обобщаются и вместе с новой дополнительной информацией поступают в распоряжение экспертов, после чего они уточняют свои первоначальные ответы. Такая процедура повторяется несколько раз до достижения приемлемой сходимости совокупности высказанных мнений. Результаты эксперимента показали приемлемую сходимость оценок экспертов после пяти туров опроса.

Метод Дельфи первоначально был предложен О. Хелмером как итеративная процедура при проведении мозговой атаки, которая должна помочь снизить влияние психологических факторов при проведении повторных заседаний и повысить объективность результатов. Однако почти одновременно Дельфи-процедуры стали основным средством повышения объективности экспертных опросов с использованием количественных оценок при оценке деревьев цели и при разработке сценариев.

Процедура Дельфи-метода:

1) в упрощённом виде организуется последовательность циклов мозговой атаки;

2) в более сложном виде разрабатывается программа последовательных индивидуальных опросов обычно с помощью вопросников,

исключая контакты между экспертами, но предусматривающая ознакомление их с мнениями друг друга между турами; вопросы от тура к туру могут уточняться;

3) в наиболее развитых методиках экспертам присваиваются весовые коэффициенты значимости их мнений, вычисляемые на основе предшествующих опросов, уточняемые от тура к туру и учитываемые при получении обобщённых результатов оценок.

Первое практическое применение метода Дельфи к решению некоторых задач Министерства обороны США во второй половине 40-х годов показало его эффективность и целесообразность распространения на широкий класс задач, связанных с оценкой будущих событий.

Исследуемые проблемы: научные открытия, рост народонаселения, автоматизация производства, освоение космоса, предотвращение войны, военная техника. Результаты статистической обработки мнений экспертов позволили нарисовать вероятную картину будущего мира в указанных шести аспектах. Была оценена также степень согласованности мнений экспертов, которая оказалась приемлемой после проведения четырёх туров опроса.

Недостатки метода Дельфи:

- значительный расход времени на проведение экспертизы, связанный с большим количеством последовательных повторений оценок;
- необходимость неоднократного пересмотра экспертом своих ответов вызывает у него отрицательную реакцию, что сказывается на результатах экспертизы.

Дальнейшим развитием метода Дельфи являются методы QUWST, SEER, PATTERN.

Методы типа дерева целей. Идея метода дерева целей впервые была предложена Черчменом в связи с проблемами принятия решений в промышленности. Термин «дерево целей» подразумевает использование иерархической структуры, полученной путём разделения общей цели на подцели, а их, в свою очередь, на более детальные составляющие – новые подцели, функции и т.д. Как правило, этот термин используется для структур, имеющих отношение строгого древовидного порядка, но метод дерева целей используется иногда и применительно к «слабым» иерархиям, в которых одна и та же вершина нижележащего уровня может быть одновременно подчинена двум или нескольким вершинам вышележащего уровня.

Древовидные иерархические структуры используются и при исследовании и совершенствовании организационных структур. Однако, не всегда разрабатываемое даже для анализа целей дерево может быть представлено в терминах целей. Иногда, например при анализе целей научных исследований, удобнее говорить о дереве направлений про-

гнозирования. В. М. Глушковым, например, был предложен и в настоящее время широко используется термин «прогнозный граф». При использовании этого понятия появляется возможность более точно определить понятие дерева как связанного ориентированного графа, не содержащего петель, каждая пара вершин которого соединяется единственной цепью.

Морфологические методы. Основная идея морфологических методов – систематически находить все «мыслимые» варианты решения проблемы или реализации системы путём комбинирования выделенных элементов или их признаков. Идеи морфологического образа мышления восходят к Аристотелю, Платону, к известной средневековой модели механизации мышления Р. Луллия. В систематизированном виде морфологический подход был разработан и применён впервые швейцарским астрономом Ф. Цвикки и долгое время был известен как метод Цвикки.

Цвикки предложил три метода морфологического исследования.

Первый метод – метод систематического покрытия поля (МСПП), основанный на выделении так называемых опорных пунктов знания в любой исследуемой области и использовании для заполнения поля некоторых сформулированных принципов мышления. Второй – метод отрицания и конструирования (МОК), базирующийся на идее Цвикки, заключающейся в том, что на пути конструктивного прогресса стоят догмы и компромиссные ограничения, которые есть смысл отрицать, и, следовательно, сформулировав некоторые предложения, полезно заменить их затем на противоположные и использовать при проведении анализа.

Третий – метод морфологического ящика (ММЯ), нашедший наиболее широкое распространение. Идея ММЯ состоит в определении всех «мыслимых» параметров, от которых может зависеть решение проблемы, и представлении их в виде матриц-строк, а затем в определении в этом морфологическом ящике (матрице) всех возможных сочетаний параметров по одному из каждой строки. Полученные таким образом варианты могут затем подвергаться оценке и анализу с целью выбора наилучшего. Морфологический ящик может быть не только двумерным. Например, А. Холл использовал для исследования структуры систем трехмерный ящик.

Морфологические ящики Цвикки нашли широкое применение для анализа и разработки прогноза в технике. Для организационных же систем, систем управления такой ящик, который, по-видимому, был бы многомерным, практически невозможно построить. Поэтому, используя идею морфологического подхода для моделирования организационных систем, разрабатывают языки моделирования или языки

проектирования, которые применяют для порождения возможных ситуаций в системе, возможных вариантов решения и часто – как вспомогательное средство формирования нижних уровней иерархической структуры как при моделировании структуры целей, так и при моделировании организационных структур. Примерами таких языков служат: системно-структурные языки (язык функций и видов структуры, номинально-структурный язык), язык ситуационного управления, языки структурно-лингвистического моделирования.

3.2. МОДЕЛИ ПРОЦЕССОВ И СИСТЕМ НА ОСНОВЕ ДЕКОМПОЗИЦИИ И АГРЕГИРОВАНИЯ

Основной операцией анализа является разделение целого на части. Задача распадается на подзадачи, системы – на подсистемы, цели – на подцели и т.д. При необходимости этот процесс повторяется, что приводит к иерархическим древовидным структурам. Обычно объект анализа сложен, слабо структурирован, плохо формализован, поэтому операцию декомпозиции выполняет эксперт. Если поручить анализ одного и того же объекта разным экспертам, то полученные древовидные списки будут различаться. Качество построенных экспертами деревьев зависит как от их компетентности в данной области знаний, так и от применяемой методики декомпозиции.

Обычно эксперт легко разделяет целое на части, но испытывает затруднения, если требуется доказательство полноты и безизбыточности предлагаемого набора частей. Стремясь перейти от чисто эвристического, интуитивного подхода к алгоритмическому выполнению декомпозиции, эксперт разделяет целое, основываясь на модели рассматриваемой системы.

Операция декомпозиции представляется как сопоставление объекта анализа с некоторой моделью, как выделение в нём того, что соответствует элементам взятой модели. Поэтому на вопрос, сколько частей должно получиться в результате декомпозиции, можно дать следующий ответ: столько, сколько элементов содержит модель, взятая в качестве основания. Вопрос о полноте декомпозиции – это вопрос завершенности модели.

Установив, что декомпозиция осуществляется с помощью некоторой модели, сквозь которую мы как бы рассматриваем расчленяемое целое, далее следует ответить на естественно возникающие вопросы: 1) модели какой системы следует брать в качестве оснований декомпозиций? 2) какие именно модели надо брать?

Всякий анализ проводится для определения того, какую систему следует рассматривать. Система, с которой связан объект анализа, и

система, по моделям которой проводится декомпозиция, не обязательно совпадают, и хотя они имеют определённое отношение друг к другу, это отношение может быть любым: одна из них может быть подсистемой или надсистемой для другой, они могут быть и разными, но как-то связанными системами.

Отметим также, что иногда в качестве оснований декомпозиции полезно не только перебирать разные модели целевой системы, но и брать сначала модели надсистемы, затем самой системы и, наконец, подсистем.

Однако чаще всего в практике системного анализа в качестве объекта декомпозиции берётся нечто, относящееся к проблемосодержащей системе и к исследуемой проблеме, а в качестве оснований декомпозиции берутся модели проблеморазрешающей системы.

Перейдём теперь к рассмотрению вопроса о том, какие модели брать за основания декомпозиции. Прежде всего, напомним, что при всём практически необозримом многообразии моделей формальных типов моделей немного: это модели «черного ящика», состава, структуры, конструкции (структурной схемы) – каждая в статическом или динамическом варианте. Это позволяет организовать нужный перебор типов моделей, полный или сокращённый, в зависимости от необходимости.

Связь между формальной и содержательной моделями. Основанием для декомпозиции может служить только конкретная, содержательная модель рассматриваемой системы. Выбор формальной модели лишь подсказывает, какого типа должна быть модель-основание; её следует наполнить содержанием, чтобы она стала основанием для декомпозиции.

Полнота декомпозиции обеспечивается полнотой модели-основания, а это означает, что следует позаботиться о полноте формальной модели. Благодаря формальности, абстрактности такой модели часто удаётся добиться её абсолютной полноты.

Пример 3.1. Формальный перечень типов ресурсов состоит из энергии, материи, времени, информации (для социальных систем добавляются кадры и финансы). При анализе ресурсного обеспечения любой конкретной системы этот перечень не даёт пропустить что-то важное.

Пример 3.2. Если в качестве модели жизненного цикла принять формулировку «всё имеет начало, середину и конец», то такая модель является формально полной. Конечно, эта модель настолько общая, что оказывается мало полезной во многих конкретных случаях. Так, при рассмотрении жизненного цикла проблем приходится использовать более детальные модели.

Итак, полнота формальной модели должна быть предметом особого внимания. Поэтому одна из важных задач информационного

обеспечения системного анализа и состоит в накоплении наборов полных формальных моделей. Полнота формальной модели является необходимым, но не достаточным условием для полноты декомпозиции. В конечном счёте, всё зависит от полноты содержательной модели, которая строится «по образу» формальной модели, но не тождественна ей.

Алгоритмизация процесса декомпозиции. Мы рассмотрели некоторые аспекты того, каким образом эксперт осуществляет единичный акт разложения целого на части. Теперь можно дать дальнейшие рекомендации по осуществлению всего многоступенчатого процесса декомпозиции, от начальной декомпозиции первого уровня до последнего уровня, завершающего данный этап анализа.

Компромиссы между полнотой и простотой. Начнём с обсуждения требований к древовидной структуре, которая получится как итог работы по всему алгоритму. С количественной стороны эти требования сводятся к двум противоречивым принципам: *полноты* (проблема должна быть рассмотрена максимально всесторонне и подробно) и *простоты* (все дерево должно быть максимально компактным – «вширь» и «вглубь»). Эти принципы относятся к количественным характеристикам (размерам) дерева. Компромиссы между ними вытекают из качественного требования – главной цели: свести сложный объект анализа к конечной совокупности простых подобъектов либо (если это не удаётся) выявить конкретную причину неустранимой сложности.

Принцип простоты требует сокращать размеры дерева. Мы уже знаем, что размеры «вширь» определяются числом элементов модели, служащей основанием декомпозиции. Поэтому принцип простоты вынуждает брать как можно более компактные модели-основания. Наоборот, принцип полноты заставляет брать как можно более развитые, подробные модели. Компромисс достигается с помощью понятия *существенности*: в модель-основание включаются только компоненты, существенные по отношению к цели анализа (релевантные). Как видим, это понятие неформальное, поэтому решение вопроса о том, что же является в данной модели существенным, а что – нет, возлагается на эксперта. Чтобы облегчить работу эксперта, в алгоритме должны быть предусмотрены возможности внесения поправок и дополнений в модель-основание. Одна из таких возможностей заключается в дополнении элементов, которые эксперт счёл существенными, ещё одним элементом «всё остальное»; он может не использоваться экспертом для декомпозиции, но будет постоянно пробуждать у эксперта сомнения в полноте предложенной им модели. Другая возможность состоит в разукрупнении, разбиении отдельных элементов модели-основания в случае необходимости, которая может возникнуть на последующих стадиях анализа.

Перейдём теперь к вопросу о размерах дерева «вглубь», т.е. о числе «этажей» дерева, числе уровней декомпозиции. Конечно, желательно, чтобы оно было небольшим (принцип простоты), но принцип полноты требует, чтобы в случае необходимости можно было продолжать декомпозицию как угодно долго до принятия решения о её прекращении по данной ветви. Такое решение принимается в нескольких случаях. Первый, к которому мы обычно стремимся, наступает, когда композиция привела к получению результата, не требующего дальнейшего разложения, т.е. результата простого, заведомо выполнимого; будем называть его *элементарным*. Для некоторых задач (например, математических, технических и т.п.) понятие элементарности может быть конкретизировано до формального признака, в других задачах анализа оно неизбежно остаётся неформальным и проверка фрагментов декомпозиции на элементарность поручается экспертам.

Неэлементарный фрагмент подлежит дальнейшей декомпозиции по другой (не использовавшейся ранее) модели-основанию. Если эксперт перебрал все модели, но не достиг элементарности на какой-то ветви дерева, то прежде всего выдвигается предположение, что дальнейшая декомпозиция может всё-таки довести анализ до получения элементарных фрагментов, и следует дать эксперту возможность продолжить декомпозицию. Такая возможность состоит во введении новых элементов в модель-основание и продолжении декомпозиции по ним. Поскольку новые существенные элементы могут быть получены только расщеплением уже имеющихся, то в алгоритме декомпозиции должна быть заложена возможность возврата к использованным ранее основаниям. При этом нет необходимости рассматривать заново все элементы модели, так как обрабатываемый фрагмент находится на ветви, соответствующей только одному элементу каждого основания. Тогда следует рассмотреть возможность расщепления именно этого элемента (например, при рассмотрении системы «вуз» вход «абитуриенты» можно разделить на абитуриентов со стажем и без него, выход «научная информация» – на выходы «монографии», «статьи», «отчёты по НИР» и т.п.). На этой же стадии можно рекомендовать эксперту решить, не настала ли пора выделить из «всего остального» и включить в число существенных ещё один элемент. Пройдя, таким образом, всю предысторию неэлементарного фрагмента, мы получаем новые основания для его декомпозиции, а значит, и возможность продолжить анализ, надеясь достичь элементарности по всем ветвям.

Итак, итеративность алгоритма декомпозиции придаёт ему вариативность, возможность пользоваться моделями различной детальности на разных ветвях, углублять детализацию.

Агрегирование, эмерджентность, внутренняя целостность систем. Операцией, противоположной декомпозиции, является операция агрегирования, т.е. объединения нескольких элементов в единое целое. Необходимость агрегирования может вызываться различными целями и сопровождаться разными обстоятельствами, что приводит к различным способам агрегирования. Однако у всех агрегатов есть одно общее свойство, получившее название *эмерджентность*. Это свойство присуще всем системам.

Эмерджентность как проявление внутренней целостности системы. Будучи объединёнными, взаимодействующие элементы образуют систему, которая обладает не только внешней целостностью, обособленностью от окружающей среды, но и внутренней целостностью, природным единством. Если внешняя целостность отображается моделью «чёрного ящика», то внутренняя целостность связана со структурой системы. Наиболее яркое проявление внутренней целостности системы состоит в том, что свойства системы не являются только суммой свойств её составных частей. Система есть нечто большее, она в целом обладает такими свойствами, которых нет ни у одной из её частей, взятой в отдельности. Модель структуры подчёркивает связанность элементов, их взаимодействие. Мы же стремимся сейчас сделать акцент на том, что при объединении частей в целое возникает нечто качественно новое, такое, чего не было и не могло быть без этого объединения.

Эмерджентность как результат агрегирования. Такое «внезапное» появление новых качеств у систем и дало основание присвоить этому их свойству название эмерджентности. Английский термин *emergence* означает возникновение из ничего, внезапное появление, неожиданную случайность. В специальной литературе на русском языке не делалось попыток найти эквивалентный русский термин.

Возникновение качественно новых свойств при агрегировании элементов есть частное, но яркое проявление всеобщего закона диалектического материализма – закона перехода количества в качество. Чем больше отличаются свойства совокупности от суммы свойств элементов, тем выше организованность системы.

3.3. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ ДИАГРАММАМИ UML

В работах, описывающих средства разработки программных продуктов, то и дело мелькает сокращение UML, которое означает *Unified Modeling Language* – Унифицированный Язык Моделирования. Знако-

миться с UML в той или иной степени придётся, поскольку UML – это стандартная нотация визуального моделирования программных систем, принятая консорциумом Object Managing Group (OMG) осенью 1997 г., и на сегодняшний день она поддерживается многими объектно-ориентированным CASE продуктами, включая Rational Rose 98i.

Визуальное моделирование. Итак, что же такое UML, и почему этому языку моделирования уделяется в последнее время столь большое внимание? Нужно ли его изучать? Как его использовать при разработке программных проектов?

Дело в том, что в последнее время наблюдается общее повышение интереса ко всем аспектам, связанным с разработкой сложных программных приложений. Для многих компаний корпоративное программное обеспечение и базы данных (БД) представляют *стратегическую ценность*. Существует высокая заинтересованность в разработке и верификации методов и подходов, позволяющих автоматизировать создание сложных программных информационных систем (ИС). Известно, что систематическое использование таких методов позволяет значительно улучшить качество, сократить стоимость и время поставки ИС. В настоящее время эти методы включают в себя:

- компонентную технологию разработки моделей ИС;
- визуальное программирование (RAD-средства);
- использование образцов (patterns) при проектировании ИС;
- визуальное представление различных аспектов проекта (визуальное моделирование, CASE-средства).

Визуальные модели широко используются в существующих технологиях управления проектированием систем, сложность, масштабы и функциональность которых постоянно возрастают. В практике эксплуатации ИС постоянно приходится решать такие задачи, как: физическое перераспределение вычислений и данных, обеспечение параллелизма вычислений, репликация БД, обеспечение безопасности доступа к ИС, оптимизация балансировки нагрузки ИС, устойчивость к сбоям и т.п.

Построение модели корпоративной ИС до её программной разработки или до начала проведения архитектурной реконструкции столь же необходимо, как наличие проектных чертежей перед строительством большого здания. Хорошие модели ИС позволяют наладить *плодотворное взаимодействие* между заказчиками, пользователями и командой разработчиков. Визуальные модели обеспечивают ясность представления выбранных архитектурных решений и позволяют понять разрабатываемую систему во всей её полноте. Сложность разрабатываемых систем продолжает увеличиваться, и поэтому возрастает

актуальность использования «хороших» методов моделирования ИС. Язык моделирования, как правило, включает в себя:

- элементы модели – фундаментальные концепции моделирования и их семантику;
- нотацию – визуальное предоставление элементов моделирования;
- принципы использования – правила применения элементов в рамках построения тех или иных типов моделей ИС.

Построение визуальных моделей позволяет решить сразу несколько типичных проблем. Во-первых, и это главное, технология визуального моделирования позволяет работать со сложными и очень сложными системами и проектами. И не важно, преобладает ли в проекте «техническая сложность» (статическая) или «динамическая сложность управления». Сложность программных систем возрастает по мере создания новых версий. И в какой-то момент наступает «эффект критической массы», когда дальнейшее развитие ИС становится невозможным, поскольку уже никто не представляет в целом «что и почему происходит». Происходит *потеря управлением проектом*. Внешней причиной или толчком возникновения этого неприятного эффекта может послужить, например, увольнение ведущего программиста или системного аналитика.

Во-вторых, визуальные модели позволяют содержательно организовывать общение между заказчиками и разработчиками. Шутка о том, что «заказчик что-то хочет, но точно не знает, чего именно», с завид-

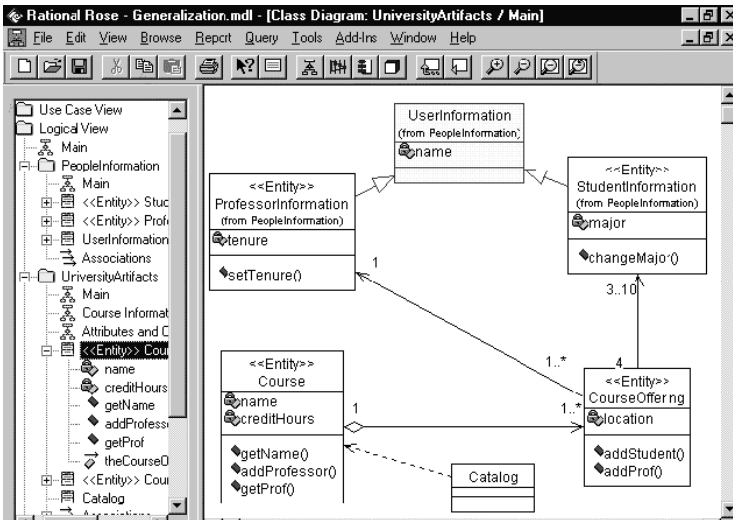


Рис. 3.1. Интерфейс Rational Rose

ным постоянством часто оказывается былью. А если на начальном этапе работы над проектом ИС заказчик думает, что точно знает, что хочет, то, как правило, и об этом свидетельствует богатый опыт, его требования изменяются («плывут») в ходе выполнения проекта. С одной стороны, аппетит приходит во время еды, а с другой, высокая динамика бизнеса объективно заставляет менять требования к разрабатываемой (или поддерживаемой) ИС.

Визуальное моделирование не является «серебряной пулей», способной раз и навсегда решить все проблемы, однако его использование существенно облегчает достижение таких целей, как: повышение качества программного продукта; сокращение стоимости проекта; поставка системы в запланированные сроки.

Как разбивать сложную систему на части? При проектировании сложной ИС её разбивают на части, каждая из которых затем рассматривается отдельно. Возможны два различных способа такого разбиения ИС на подсистемы: структурное (или функциональное) разбиение и объектная (компонентная) декомпозиция.

Суть функционального разбиения хорошо отражена в известной формуле:

Программа = Данные + Алгоритмы.

При функциональном разбиении программной системы её структура может быть описана блок-схемами, узлы которых представляют собой «обрабатывающие центры» (функции), а связи между узлами описывают движение данных.

Объектная декомпозиция в последнее время называется *компонентной*, что нашло отражение в специальном термине: «разработка, основанная на компонентах» (Component Based Development – CBD). При этом используется иной принцип декомпозиции – система разбивается на «активные сущности» – объекты или компоненты, которые *взаимодействуют* друг с другом, *обмениваясь сообщениями* и выступая друг к другу в отношении «клиент/сервер». Сообщения, которые может принимать объект, определены в его интерфейсе. В этом смысле посылка сообщения «объекту-серверу» эквивалентна вызову соответствующего метода объекта.

Так вот, если при проектировании информационная система разбивается на объекты (компоненты), то UML может быть использован для её визуального моделирования. Если используется функциональная декомпозиция ИС, то UML не нужен, и следует использовать другие (структурные) нотации.

Отдельная тема для обсуждения – нужно ли программировать в объектах (компонентах)? Споры на эту тему относятся к разряду «ре-

лигиозных войн». Есть убеждённые сторонники в обоих лагерях. Уместно заметить, что все современные RAD-средства программирования используют библиотеки компонент, позволяющие *повторно* использовать отлаженный программный код, что значительно облегчает сборку программных приложений. Такое «сборочное программирование» стало возможным за счёт использования объектов и привело к изменению квалификационной оценки программистов за рубежом – «программист – это тот, кто умеет программировать в компонентах», т.е. это не «писатель программного кода», как принято считать у нас.

С точки зрения визуального моделирования, UML можно охарактеризовать следующим образом. UML предоставляет выразительные средства для создания визуальных моделей, которые единообразно понимаются всеми разработчиками, вовлечёнными в проект, и являются средством коммуникации в рамках проекта.

Унифицированный Язык Моделирования (UML):

- не зависит от объектно-ориентированных (ОО) языков программирования;
- не зависит от используемой методологии разработки проекта;
- может поддерживать любой ОО язык программирования.

UML является *открытым* и обладает средствами расширения базового ядра. На UML можно содержательно описывать классы, объекты и компоненты в различных предметных областях, часто сильно отличающихся друг от друга.

Как создавался UML? В середине 90-х существовало более 50 различных объектно-ориентированных языков моделирования. И разработчики, и заказчики испытывали беспокойство при выборе метода проектирования ИС, который, как правило, включал в себя и собственную нотацию. В это время стали появляться новые версии таких распространенных методов, как: Booch'93, OMT-2 (Object Modelling Technique), Fusion, OOSE (Object-Oriented Software Engineering). Возникла насущная потребность в стандартизации и унификации.

Разработка UML была начата в октябре 1994 Грэди Бучем (Grady Booch) и Джимом Рамбо (Jim Rumbaugh) в Rational Software Corporation как унификация двух методов: Booch'93 и OMT. Первая версия Унифицированного Метода (Unified Method 0.8) была опубликована в октябре 1995. Осенью 1995 г. к работе присоединился Айвер Якобсон (Ivar Jacobson), включив в процесс унификации свой метод OOSE. Таким образом, на первом концептуальном этапе UML имел трёх авторов: Буча, Рамбо и Якобсона, каждый из которых являлся идеологом своего ОО метода визуального моделирования.

В октябре 1996 года была выпущена редакция UML 0.91, в которой отражены многочисленные пожелания, полученные в течение

1996 года. В это же время выяснилось, что ряд влиятельных организаций, связанных с компьютерным бизнесом, стал рассматривать UML как стратегический элемент своей деятельности. Катализатором объединения усилий по унификации UML стал выпуск консорциумом OMG (Object Management Group) «запроса на предложения» по UML (RFP – Request for Proposal). Известно, что выпуск RFP является первым шагом процедуры принятия OMG того или иного стандарта. После этого Rational Software под своей эгидой создала организацию «Консорциум UML партнеров» («UML Partnersconsortium») для разработки формального определения UML 1.0 как стандарта. В работе консорциума приняли участие представители таких известных компаний, как: Digital Equipment Corp., Hewlett-Packard, i-Logix, Intelli Corp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI, Unisys.

В результате в январе 1997 г. в OMG был представлен вариант UML 1.0. как первый RFP-отклик. В это же время второй RFP-отклик независимо от «UML Partnersconsortium» представили такие организации, как: Objec Time; Platinum Technology; Ptech; Taskon & Reich Technologies и Softeam. Для объединения предложений по двум представленным проектам UML эти компании также присоединились к «UML Partnersconsortium», и в результате был подготовлен вариант UML 1.1. Именно этот вариант в ноябре 1997 года был утверждён как стандарт. Формальное описание UML 1.1 в виде семи pdf-файлов можно найти, например, на сайтах OMG, Rational Software, Platinum Technology. По заголовкам можно судить о составе официальной документации по UML:

1. UML Summary,
2. UML Notation Guide,
3. UML Semantics,
4. UML OCL (Object Constraint Language Specification),
5. UML Objectory (UML Extension for Objectory Process for Software Engineering),
6. UML Business (UML Extension for Business Modeling),
7. UML Metamodel_Diagram.

Диаграмма вариантов использования. При моделировании информационных систем с помощью UML начинают с диаграммы вариантов использования, поскольку она (диаграмма) позволяет описать функциональные и поведенческие требования к системе. Важность этой диаграммы сложно переоценить, поскольку именно здесь формулируются требования к системе со стороны заказчика, то, что заказчик хочет видеть, получив систему.

В диаграмме вариантов использования применяются следующие обозначения.

- Действующее лицо (actor) – это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. Несмотря на то, что на диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок, действующее лицо может также быть внешней системой, которой необходима некоторая информация от данной системы. Показывать на диаграмме действующих лиц следует только в том случае, когда им действительно необходимы некоторые варианты использования.

- Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать.

- Связь – коммуникация между элементами диаграммы.

- Прямая связь – коммуникация, показывающая инициатора связи, между элементами диаграммы.

- Связь включения или расширения – применяется при описании изменений в нормальном поведении системы. Она позволяет варианту использования обязательно или только при необходимости использовать функциональные возможности другого соответственно. Для обозначения типа связи над стрелочкой, с помощью комментария, пишется <<include>> (включение) или <<extend>> (расширение).

- Связь включения – применяется в тех ситуациях, когда имеется какой-либо фрагмент поведения системы, который повторяется более чем в одном варианте использования. С помощью таких связей обычно моделируют многократно используемую функциональность.

Описание (комментарий) обычно содержит следующие данные:

- краткое описание;

- предусловия (pre-conditions);

- основной поток событий;

- альтернативный поток событий (или несколько альтернативных потоков);

- постусловия (post-conditions).

Пример диаграммы вариантов использования для информационной системы ресторана представлен на рис. 3.2.

Пользователи информационной системы следует разделить на две категории: пользователь, т.е. гость ресторана, который может

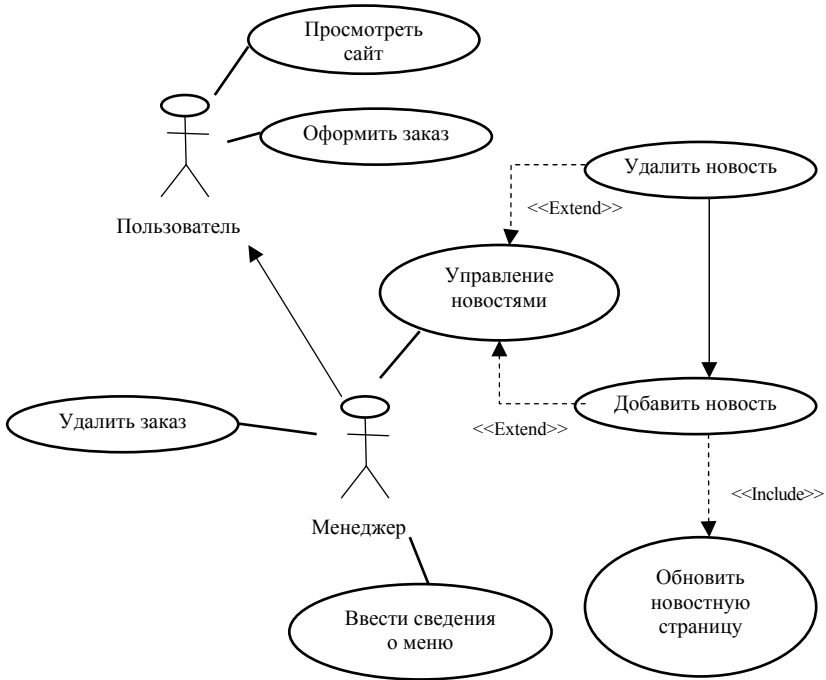


Рис. 3.2. Диаграмма вариантов использования

просматривать сайт и оформляет заказ; а также менеджер, который помимо функций пользователя может вводить в систему сведения о меню, управляет новостями, также может оформлять заказ либо при необходимости удалить заказ из системы.

Разрабатывая диаграммы вариантов использования, старайтесь придерживаться следующих правил:

- Не моделируйте связи между действующими лицами. По определению действующие лица находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к её компетенции.

- Не пытайтесь отобразить «алгоритм» работы системы, диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения. Для отображения порядка выполнения вариантов использования применяют диаграммы деятельности.

Диаграмма деятельности. Если в диаграмме вариантов использования мы ставим перед собой цель, к которой идём, то в диаграмме деятельности показывается путь, по которому мы будем идти.

Диаграмма деятельности – это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Результат может привести к изменению состояния системы или возвращению некоторого значения. Диаграмма деятельности отличается от традиционной блок-схемы:

- 1) более высоким уровнем абстракции;
- 2) возможностью представления с помощью диаграмм деятельности управления параллельными потоками наряду с последовательным управлением.

Одно из основных направлений использования диаграмм деятельности – отображение внутрисистемной точки зрения на прецедент. Диаграммы деятельности применяют для описания шагов, которые должна предпринять система после того, как инициирован прецедент.

Разработка диаграммы деятельности преследует цели:

1. Детализировать особенности алгоритмической и логической реализации прецедентов;
2. Выделить последовательные и параллельные потоки управления;
3. Подготовить детальную документацию для взаимодействия разработчиков системы с её заказчиками и проектировщиками.

В диаграмме деятельности предусмотрены следующие обозначения.

Состояние деятельности – это продолжающийся во времени неатомарный шаг вычислений в автомате. Состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, вследствие чего выполняемую деятельность можно представить с помощью других диаграмм деятельности. Состояния деятельности не являются атомарными, т.е. могут быть прерваны. Предполагается, что для их завершения требуется заметное время.

Состояние действия – состояние, которое представляет вычисление атомарного действия, как правило – вызов операции. Состояния действия не могут быть подвергнуты декомпозиции. Они атомарны, т.е. внутри них могут происходить различные события, но выполняемая в состоянии действия работа не может быть прервана. Обычно предполагается, что длительность одного состояния действия занимает неощутимо малое время. Действие может заключаться в вызове другой операции, послышке сигнала, создании или уничтожении объекта либо в простом вычислении – скажем, значения выражения.

Состояния деятельности и состояния действия имеют одинаковое стандартное графическое обозначение – прямоугольник с закругленными краями. Внутри такого символа записывают произвольное выражение, которое должно быть уникальным в пределах одной диаграммы деятельности.

Начальное и конечное состояния на диаграммах деятельности изображаются как закрашенный кружок и закрашенный кружок внутри окружности, соответственно.

Переход – отношение между двумя состояниями, показывающее, что объект, находящийся в первом состоянии, должен выполнить некоторые действия и перейти во второе состояние. Когда действие или деятельность в некотором состоянии завершается, поток управления сразу переходит в следующее состояние действия или деятельности. Для описания этого потока и используются переходы, показывающие путь из одного состояния действия или деятельности в другое. В UML переход представляется простой линией со стрелкой.

Ветвления. Простые последовательные переходы встречаются наиболее часто, но их одних недостаточно для моделирования любого потока управления. Как и в блок-схему, в диаграмму деятельности может быть включено ветвление или множественный переход со сторожевыми условиями. Ветвление описывает различные пути выполнения в зависимости от значения некоторого булевого выражения. Графически точка ветвления представляется ромбом. В точку ветвления может входить ровно один переход, а выходить – два или более. Для каждого исходящего перехода задаётся булево выражение, которое вычисляется только один раз при входе в точку ветвления. Ни для каких двух исходящих переходов сторожевые условия не должны одновременно принимать значение «истина», иначе поток управления окажется неоднозначным. Но эти условия должны покрывать все возможные варианты, иначе поток остановится.

Разделения и слияния. Простые и ветвящиеся последовательные переходы в диаграммах деятельности используются чаще всего. Однако часто возникает потребность изображения параллельных потоков, и это особенно характерно для моделирования бизнес-процессов. В UML для обозначения разделения и слияния таких параллельных потоков выполнения используется синхронизационная черта, которая рисуется в виде жирной вертикальной или горизонтальной линии. При этом разделение имеет один входящий переход и несколько выходящих, слияние, наоборот, имеет несколько входящих переходов и один выходящий (см. рис. 3.3).

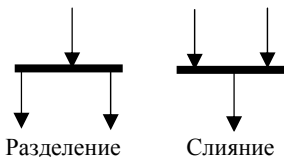


Рис. 3.3. Разделение и слияние потоков

Следует помнить, что под параллельными потоками управления имеется в виду не только истинный параллелизм, т.е. одновременное выполнение, но и последовательное выполнение с переключением между потоками, что даёт лишь

иллюзию истинного параллелизма, а также независимое прохождение потоков в произвольном порядке.

Дорожки. При моделировании течения бизнес-процессов иногда бывает полезно разбить состояния деятельности на диаграммах деятельности на группы, каждая из которых представляет отдел компании, отвечающий за ту или иную работу. В UML такие группы называются дорожками, поскольку визуально каждая группа отделяется от соседних вертикальной чертой, как плавательные дорожки в бассейне. Дорожки – это разновидность пакетов, описывающих связанную совокупность работ.

Каждой присутствующей на диаграмме дорожке присваивается уникальное имя. Никакой глубокой семантики дорожка не несёт, разве что может отражать некоторую сущность реального мира. Каждая дорожка представляет сферу ответственности за часть всей работы, изображённой на диаграмме. На диаграмме деятельности, разбитой на дорожки, каждая деятельность принадлежит ровно одной дорожке, но переходы могут пересекать границы дорожек.

Имеется некоторая связь между дорожками и параллельными потоками выполнения. Концептуально деятельность внутри каждой дорожки обычно – но не всегда – рассматривается отдельно от деятельности в соседних дорожках. Это разумно, поскольку в реальном мире подразделения организации, представленные дорожками, как правило, независимы и функционируют параллельно. Пример диаграммы деятельности представлен на рис. 3.4.

Диаграмма классов. После построения диаграммы вариантов использования и диаграммы деятельности мы получаем значительную часть функционального описания системы. Зная, какие деятельности будет выполнять система и как они будут работать, необходимо определиться со статическими данными, необходимыми для функционирования проектируемой информационной системы. Для этого используется диаграмма классов, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними.

Существует несколько подходов к созданию диаграммы классов:

- 1) с точки зрения спецификации (т.е. как вспомогательный элемент, для понимания происходящих процессов);
- 2) с точки зрения реализации (т.е. для непосредственной генерации программного кода).

Нас интересует первый подход к построению диаграммы классов, суть его заключается в нахождении необходимых и достаточных атрибутов и связей между множеством предметов реального мира, связанных общностью структуры и поведением.

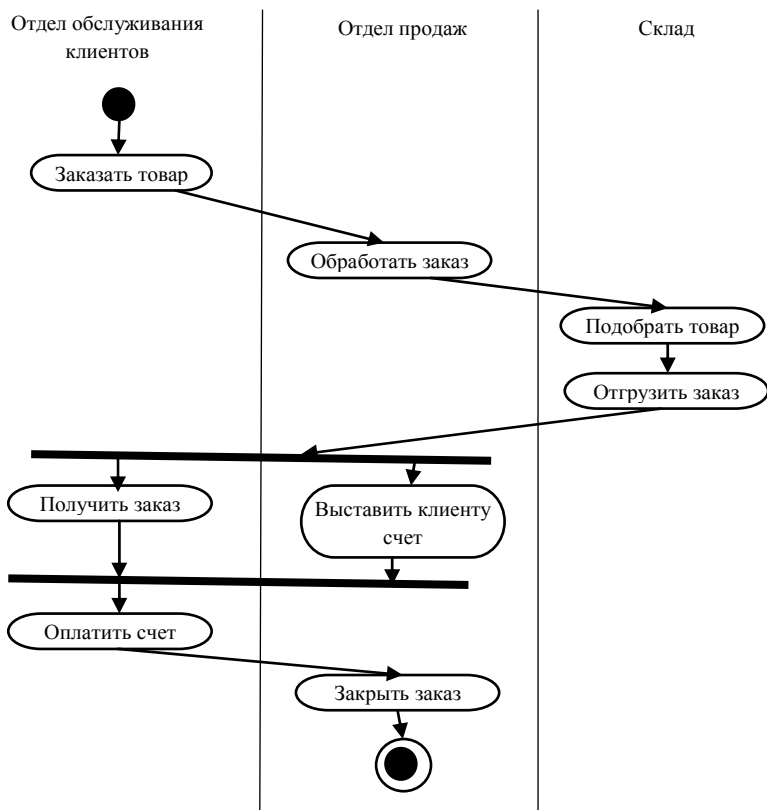


Рис. 3.4. Пример диаграммы деятельности

Атрибут – это элемент информации, связанный с классом. Например, у класса Товар могут быть атрибуты Название, Цена и Описание.

Так как атрибуты содержатся внутри класса, они скрыты от других классов. В связи с этим может понадобиться указать, какие классы имеют право читать и изменять атрибуты. Это свойство называется видимостью атрибута (attributevisibility).

У атрибута можно определить три возможных значения этого параметра. Пусть у нас имеется класс Товар с атрибутом Цена и класс Скидка, тогда тип видимости атрибута:

- Public (общий, открытый). Это значение видимости предполагает, что атрибут будет виден всеми остальными классами. Любой класс может просмотреть или изменить значение атрибута. В таком слу-

чае класс Скидка может изменить значение атрибута Цена класса Товар. В соответствии с нотацией UML общему атрибуту предшествует знак «+».

– Private (закрытый, секретный). Соответствующий атрибут не виден никаким другим классом. Класс Товар будет знать значение атрибута Цена и сможет изменять его, но класс Скидка не сможет его ни увидеть, ни редактировать. Если это понадобится, он должен попросить класс Товар просмотреть или изменить значение этого атрибута, что обычно делается с помощью общих операций. Закрытый атрибут обозначается знаком «-» в соответствии с нотацией UML.

– Protected (защищённый). Такой атрибут доступен только самому классу и его потомкам. Допустим, что у нас имеется два различных типа товаров – Ноутбуки и Планшетки. Таким образом, мы получаем два других класса Ноутбуки и Планшетки, являющихся потомками класса Товар. Защищённый атрибут Цена можно просмотреть или изменить из классов Товар, Ноутбуки и Планшетки, но не из класса Скидка. Нотация UML для защищённого атрибута – это знак «#». Следует помнить, что имена классов должны быть уникальными в рамках проекта, а также на диаграммах классов изображаются атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.

Механизм, позволяющий разделять классы на категории – Стереотипы. В языке UML определены три основных стереотипа классов: Boundary (граница), Entity (сущность) и Control (управление).

Граничные классы. Граничными классами называются такие классы, которые расположены на границе системы и всей окружающей среды. Это экранные формы, отчёты, интерфейсы с аппаратурой (такой как принтеры или сканеры) и интерфейсы с другими системами. Чтобы найти граничные классы, надо исследовать диаграммы вариантов использования. Каждому взаимодействию между действующим лицом и вариантом использования должен соответствовать, по крайней мере, один граничный класс. Именно такой класс позволяет действующему лицу взаимодействовать с системой.

Классы-сущности. Классы-сущности содержат хранимую информацию. Они имеют наибольшее значение для пользователя, и потому в их названиях часто используют термины из предметной области. Обычно для каждого класса-сущности создают таблицу в базе данных.

Управляющие классы. Управляющие классы отвечают за координацию действий других классов. Обычно у каждого варианта использования имеется один управляющий класс, контролирующий последо-

вательность событий этого варианта использования. Управляющий класс отвечает за координацию, но сам не несёт в себе никакой функциональности, так как остальные классы не посылают ему большого количества сообщений. Вместо этого он сам посылает множество сообщений. Управляющий класс просто делегирует ответственность другим классам, по этой причине его часто называют классом-менеджером.

Связь представляет собой семантическую взаимосвязь между классами. Она даёт классу возможность узнавать об атрибутах, операциях и связях другого класса. Иными словами, чтобы один класс мог послать сообщение другому на диаграмме последовательности или кооперативной диаграмме, между ними должна существовать связь.

Существуют четыре типа связей, которые могут быть установлены между классами: ассоциации, зависимости, агрегации и обобщения.

Ассоциация (association) – это семантическая связь между классами. Их рисуют на диаграмме классов в виде обыкновенной линии.

Ассоциации могут быть двунаправленными или однонаправленными. На языке UML двунаправленные ассоциации рисуют в виде простой линии без стрелок или со стрелками с обеих её сторон. На однонаправленной ассоциации изображают только одну стрелку, показывающую её направление.

Направление ассоциации можно определить, изучая диаграммы последовательности и кооперативные диаграммы. Если все сообщения на них отправляются только одним классом и принимаются только другим классом, но не наоборот, между этими классами имеет место однонаправленная связь. Если хотя бы одно сообщение отправляется в обратную сторону, ассоциация должна быть двунаправленной.

Ассоциации могут быть рефлексивными. Рефлексивная ассоциация предполагает, что один экземпляр класса взаимодействует с другими экземплярами этого же класса.

Зависимости. *Связи зависимости* (dependency) также отражают связь между классами, но они всегда однонаправлены и показывают, что один класс зависит от определений, сделанных в другом. Зависимости изображают в виде стрелки, проведённой пунктирной линией.

Агрегации (aggregations) представляют собой более тесную форму ассоциации. Агрегация – это связь между целым и его частью. Например, у вас может быть класс Автомобиль, а также классы Двигатель, Покрышки и классы для других частей автомобиля. В результате объект класса Автомобиль будет состоять из объекта класса Двигатель, четырёх объектов Покрышек и т.д. Агрегации визуализируют в виде линии с ромбиком у класса, являющегося целым.

В дополнение к простой агрегации UML вводит более сильную разновидность агрегации, называемую композицией. Согласно композиции, объект-часть может принадлежать только единственному целому, и, кроме того, как правило, жизненный цикл частей совпадает с циклом целого: они живут и умирают вместе с ним. Любое удаление целого распространяется на его части.

Такое каскадное удаление нередко рассматривается как часть определения агрегации, однако оно всегда подразумевается в том случае, когда множественность роли составляет 1..1; например, если необходимо удалить Клиента, то это удаление должно распространиться и на Заказы (и, в свою очередь, на Строки заказа).

Обобщения. С помощью обобщений (generalization) показывают связи наследования между двумя классами. Большинство объектно-ориентированных языков непосредственно поддерживают концепцию наследования. Она позволяет одному классу наследовать все атрибуты, операции и связи другого. На языке UML связи наследования называют обобщениями и изображают в виде стрелок от класса-потомка к классу-предку.

Пример диаграммы классов представлен на рис. 3.5.

Типичные ошибки при создании диаграммы классов:

- 1) дублирование функций и атрибутов в классах со связью обобщение;
- 2) путаница со связями агрегации и композиции.

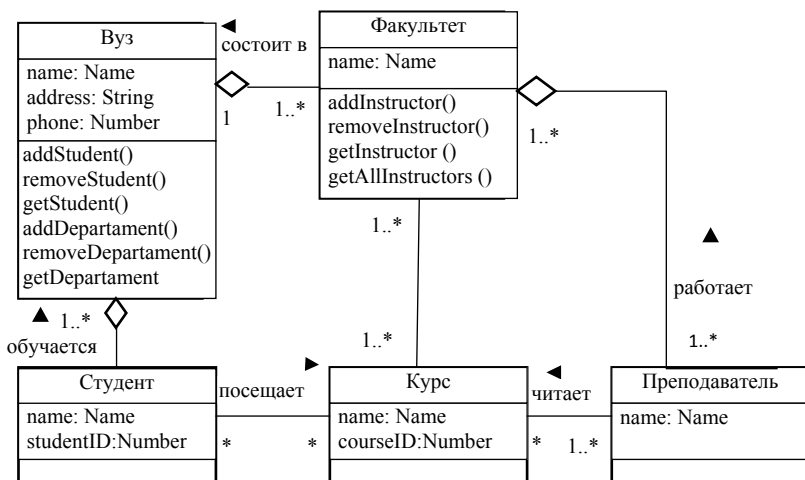


Рис. 3.5. Пример диаграммы классов

3.4. ПРОЦЕССНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Изучение любой системы предполагает создание модели системы, позволяющей произвести анализ и предсказать её поведение в определенном диапазоне условий, решать задачи анализа и синтеза реальной системы. В зависимости от целей и задач моделирование может проводиться на различных уровнях абстракции.

Модель – описание системы, отражающее определённую группу её свойств.

Описание системы целесообразно начинать с трёх точек зрения: функциональной, морфологической и информационной.

Всякий объект характеризуется результатами своего существования, местом, которое он занимает среди других объектов, ролью, которую он играет в среде. Функциональное описание необходимо для того, чтобы осознать важность системы, определить её место, оценить отношения с другими системами.

Функциональное описание (функциональная модель) должно создать правильную ориентацию в отношении внешних связей системы, её контактов с окружающим миром, направлениях её возможного изменения.

Функциональное описание исходит из того, что всякая система выполняет некоторые функции: просто пассивно существует, служит областью обитания других систем, обслуживает системы более высокого порядка, служит средством для создания более совершенных систем.

Во многом оценка функций системы (в абсолютном смысле) зависит от точки зрения того, кто её оценивает (или системы, её оценивающей).

Функционирование системы может описываться числовым функционалом, зависящим от функций, описывающих внутренние процессы системы, либо качественным функционалом (упорядочение в терминах «лучше», «хуже», «больше», «меньше» и т.д.)

Функционал, количественно или качественно описывающий деятельность системы, называют функционалом эффективности.

Функциональная организация может быть описана:

- алгоритмически;
- аналитически;
- графически;
- таблично;
- посредством временных диаграмм функционирования;
- вербально (словесно).

Описание должно соответствовать концепции развития систем определённого класса и удовлетворять некоторым требованиям:

- должно быть открытым и допускать возможность расширения (сужения) спектра функций, реализуемых системой;
- предусматривать возможность перехода от одного уровня рассмотрения к другому, т.е. обеспечивать построение виртуальных моделей систем любого уровня.

При описании системы будем рассматривать её как структуру, в которую в определённые моменты времени вводится нечто (вещество, энергия, информация) и из которой в определённые моменты времени нечто выводится.

Диаграммы потоков данных (DataFlowDiagram, DFD) – методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель таких средств – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Информационная система принимает извне потоки данных. Для обозначения элементов среды функционирования системы используется понятие внешней сущности. Внутри системы существуют процессы преобразования информации, порождающие новые потоки данных. Потоки данных могут поступать на вход к другим процессам, помещаться (и извлекаться) в накопители данных, передаваться к внешним сущностям.

Модель DFD, как и большинство других структурных моделей – иерархическая модель. Каждый процесс может быть подвергнут декомпозиции, т.е. разбиению на структурные составляющие, отношения между которыми в той же нотации могут быть показаны на отдельной диаграмме. Когда достигнута требуемая глубина декомпозиции, то процесс нижнего уровня сопровождается мини-спецификацией (текстовым описанием).

Кроме того, нотация DFD поддерживает понятие подсистемы – структурной компоненты разрабатываемой системы.

Нотация DFD – удобное средство для формирования контекстной диаграммы, т.е. диаграммы, показывающей разрабатываемую АИС в коммуникации с внешней средой. Это диаграмма верхнего уровня в иерархии диаграмм DFD. Её назначение – ограничить рамки системы, определить, где заканчивается разрабатываемая система и начинается среда.

Основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы и подсистемы;
- процессы;
- накопители данных;
- потоки данных.

Рассмотрим их подробнее.

Внешняя сущность – материальный предмет или физическое лицо, представляющее собой источник или приёмник информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой ИС, если это необходимо, или, наоборот, часть процессов ИС может быть вынесена за пределы диаграммы и представлена как внешняя сущность.



Рис. 3.6. Внешняя сущность

Внешняя сущность обозначается квадратом (рис. 3.6), расположенным как бы «над» диаграммой и бросающим на неё тень, для того, чтобы можно было выделить этот символ среди других обозначений:

При построении модели сложной ИС она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого либо может быть декомпозирована на ряд подсистем.

Подсистема (или система) на контекстной диаграмме изображается так, как показано на рис. 3.7.

Номер подсистемы служит для её идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

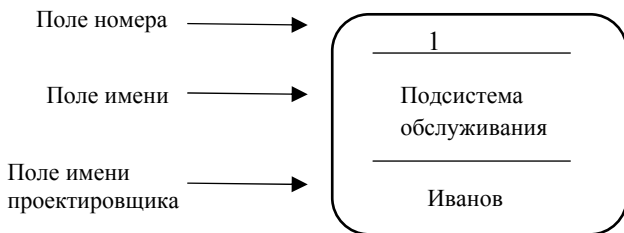


Рис. 3.7. Подсистема на диаграмме DFD

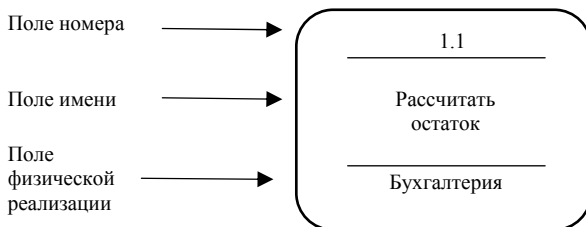


Рис. 3.8. Процесс на диаграмме DFD

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определённым алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчётов, программа, аппаратно реализованное логическое устройство и т.д.

Процесс на диаграмме потоков данных изображается, как показано на рис. 3.8.

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределённой форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- «Ввести сведения о клиентах»;
- «Выдать информацию о текущих расходах»;
- «Проверить кредитоспособность клиента».

Использование таких глаголов, как «обработать», «модернизировать» или «отредактировать» означает, как правило, недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

Накопитель данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причём способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных изображается, как показано на рис. 3.9.

D1	Получаемые счета
----	------------------

Накопитель данных идентифицируется буквой «D» и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Рис. 3.9. Накопитель данных

Накопитель данных в общем случае является прообразом будущей базы данных, и описание хранящихся в нём данных должно быть увязано с информационной моделью.

Поток данных определяет информацию, передаваемую через некоторое соединение от источника к приёмнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока (рис. 3.10). Каждый поток данных имеет имя, отражающее его содержание.

Контекстная диаграмма и детализация процессов.

Декомпозиция DFD осуществляется на основе процессов: каждый процесс может раскрываться с помощью DFD нижнего уровня.

Важную специфическую роль в модели играет специальный вид DFD – контекстная диаграмма, моделирующая систему наиболее общим образом. Контекстная диаграмма отражает интерфейс системы с внешним миром, а именно, информационные потоки между системой и внешними сущностями, с которыми она должна быть связана. Она идентифицирует эти внешние сущности, а также, как правило, единственный процесс, отражающий главную цель или природу системы насколько это возможно. И хотя контекстная диаграмма выглядит тривиальной, несомненная её полезность заключается в том, что она устанавливает границы анализируемой системы. Каждый проект должен иметь ровно одну контекстную диаграмму, при этом нет необходимости в нумерации единственного её процесса.

DFD первого уровня строится как декомпозиция процесса, который присутствует на контекстной диаграмме.

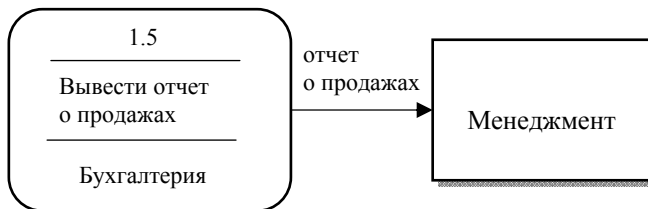


Рис. 3.10. Поток данных на диаграмме DFD

Построенная диаграмма первого уровня также имеет множество процессов, которые в свою очередь могут быть декомпозированы в DFD нижнего уровня. Таким образом строится иерархия DFD с контекстной диаграммой в корне дерева. Этот процесс декомпозиции продолжается до тех пор, пока процессы могут быть эффективно описаны с помощью коротких (до одной страницы) мини-спецификаций обработки (спецификаций процессов).

При таком построении иерархии DFD каждый процесс более низкого уровня необходимо соотнести с процессом верхнего уровня. Обычно для этой цели используются структурированные номера процессов. Так, например, если мы детализируем процесс номер 2 на диаграмме первого уровня, раскрывая его с помощью DFD, содержащей три процесса, то их номера будут иметь следующий вид: 2.1, 2.2 и 2.3. При необходимости можно перейти на следующий уровень, т.е. для процесса 2.2 получим 2.2.1, 2.2.2 и т.д.

Пример. Главная цель построения иерархического множества DFD заключается в том, чтобы сделать требования ясными и понятными на каждом уровне детализации, а также разбить эти требования на части с точно определёнными отношениями между ними. Для достижения этого целесообразно пользоваться следующими рекомендациями:

1. Размещать на каждой диаграмме от 3 до 6-7 процессов. Верхняя граница соответствует человеческим возможностям одновременного восприятия и понимания структуры сложной системы с множеством внутренних связей, нижняя граница выбрана по соображениям здравого смысла: нет необходимости детализировать процесс диаграммой, содержащей всего один или два процесса.

2. Не загромождать диаграммы несущественными на данном уровне деталями.

3. Декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов; эти две работы должны выполняться одновременно, а не одна после завершения другой.

4. Выбирать ясные, отражающие суть дела, имена процессов и потоков для облегчения понимания диаграмм, при этом стараться не использовать аббревиатуры.

5. Однократно определять функционально идентичные процессы на самом верхнем уровне, где такой процесс необходим, и ссылаться к нему на нижних уровнях.

6. Пользоваться простейшими диаграммными техниками: если что-либо возможно описать с помощью DFD, то это и необходимо делать, а не использовать для описания более сложные объекты.

7. Отделять управляющие структуры от обрабатывающих структур (т.е. процессов), локализовать управляющие структуры.

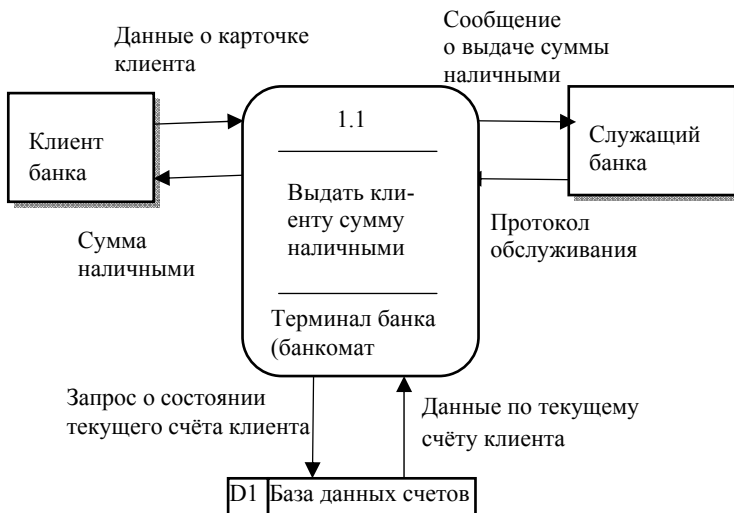


Рис. 3.11. Типы дуг в DFD

Диаграмма DFD на основе примера процесса получения некоторой суммы наличными по кредитной карточке рассмотрена на рис. 3.11.

Функциональное моделирование с помощью IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (Workflow).

IDEF0 как стандарт был разработан в 1981 году департаментом Военно-Воздушных Сил США в рамках программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Набор стандартов IDEF унаследовал своё название от этой программы (IDEF=ICAM DEFinition). В процессе практической реализации участники программы ICAM столкнулись с необходимостью разработки новых методов анализа процессов взаимодействия в промышленных системах. При этом, кроме усовершенствованного набора функций для описания бизнес-процессов, одним из требований к новому стандарту было наличие эффективной методологии взаимодействия в рамках «аналитик-специалист». Другими словами, новый метод должен был обеспечить групповую работу над созданием модели с непосредственным участием всех аналитиков и специалистов, занятых в рамках проекта.

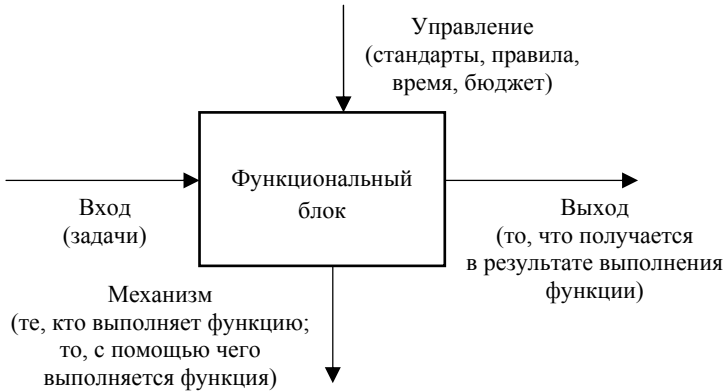


Рис. 3.12. Типы в диаграмме IDEF0

В результате поиска соответствующих решений родилась методология функционального моделирования IDEF0. С 1981 года стандарт IDEF0 претерпел несколько незначительных изменений, в основном, ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологичам США (NIST).

Основными элементами на IDEF0-диаграммах являются блоки и дуги (рис. 3.12).

Блоки служат для отображения функций (действий), выполняемых моделируемой системой. Сформулированные функции должны содержать глагольный оборот.

Например: обрабатывать деталь на станке, передать документы в отдел, разработать план-график проведения анализа, опубликовать материалы и т.д.

Дуги служат для отображения информации или материальных объектов, которые необходимы для выполнения функции или появляются в результате её выполнения (объекты, обрабатываемые системой). Под объектами в рамках функционального моделирования могут пониматься документы, физические материалы, инструменты, станки, информация, организации и даже системы.

Место соединения дуги с блоком определяет тип интерфейса.

Управляющие выполнением функции данные входят в блок сверху, в то время как информация, которая подвергается воздействию функции, показана с левой стороны блока; результаты выхода показаны с правой стороны.

Механизм (человек или автоматизированная система), который осуществляет функцию, представляется дугой, входящей в блок снизу.

Функциональный блок преобразует входную информацию (данные, материалы, средства, задачи, цели и др.) в выходную (что требуется получить в результате выполнения данной функции). Управление определяет, когда и как это преобразование может или должно произойти. Механизм (или исполнители) непосредственно осуществляет это преобразование.

За каждой дугой закрепляется замечание, которое отображает суть информации или объекта. Замечание формулируется в виде оборота существительного, отвечающего на вопрос: «Что?».

Функциональные блоки на диаграмме изображаются в виде прямоугольников, внутри которых записываются имя функции и номер блока (в правом нижнем углу прямоугольника).

Блоки располагаются на диаграмме согласно их степени важности (по мнению автора модели). При этом доминирующим является тот блок, выполнение функции которого оказывает влияние на выполнение всех остальных функций, представленных на диаграмме. К примеру, это может быть блок, содержащий контролируемую или планируемую функцию, выходы которого являются управляющими для всех остальных функциональных блоков диаграммы.

Доминирующий блок помещается, как правило, в верхнем левом углу листа диаграммы, а наименее важный блок – в правом нижнем углу. Таким образом, ступенчатость блоков на диаграмме отражает мнение автора о доминировании одних блоков относительно других.

Очень важно помнить, что доминирование блоков на диаграмме не задаёт четкой временной зависимости операций.

Стороны блока также имеют определенное значение. К левой границе блока присоединяются входные дуги, к верхней – управляющие дуги, к правой – выходные дуги, а к нижней – дуги механизмов.

Дуги на IDEF0 диаграмме изображаются в виде стрелок.

При IDEF0 моделировании используются пять типов взаимосвязей между блоками для описания их отношений.

– Взаимосвязь по управлению, когда выход одного блока влияет (является управляющей) на выполнение функции в другом блоке.

– Взаимосвязь по входу, когда выход одного блока является входом для другого.

– Обратная связь по управлению, когда выходы из одной функции влияют на выполнение других функций, выполнение которых в свою очередь влияет на выполнение исходной функции.

– Обратная связь по входу, когда выход из одной функции является входом для другой функции, выход которой является для него входом.

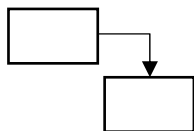


Рис. 3.13. Взаимосвязь по управлению

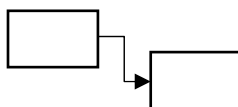


Рис. 3.14. Взаимосвязь по входу

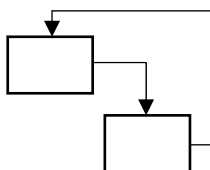


Рис. 3.15. Обратная связь по управлению

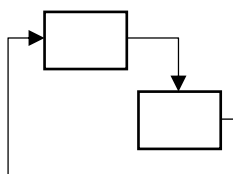


Рис. 3.16. Обратная связь по входу

– Взаимосвязь «выход-механизм», когда выход одной функции является механизмом для другой. Иначе говоря, выходная дуга одного блока является дугой механизма для другого. Такой тип связи встречается редко и относится чаще всего к подготовительным операциям.

Поскольку содержание IDEF0 диаграмм уточняется в ходе моделирования постепенно, дуги на диаграммах редко изображают один объект. Чаще всего они отображают определённый набор объектов и могут иметь множество начальных точек (источников) и определённое количество конечных точек (приёмников). В ходе разработки графической диаграммы для отражения этой особенности используют механизм разветвления/слияния дуг. Это позволяет не только уточнить с использованием замечаний содержание каждой ветви разветвлённой дуги (потока объектов), но и более точно описать, из каких наборов объектов состоит входящая в функциональный блок дуга, если она получена путём слияния.

Диаграмма декомпозиции IDEF3 – способ описания процессов с использованием структурированного метода, позволяющего эксперту в предметной области представить положение вещей как упорядоченную последовательность событий с одновременным описанием объектов, имеющих непосредственное отношение к процессу.

IDEF3 является технологией, хорошо приспособленной для сбора данных, требующихся для проведения структурного анализа системы.

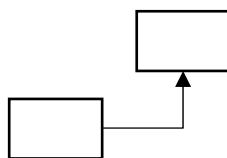


Рис. 3.17. Взаимосвязь «выход-механизм»

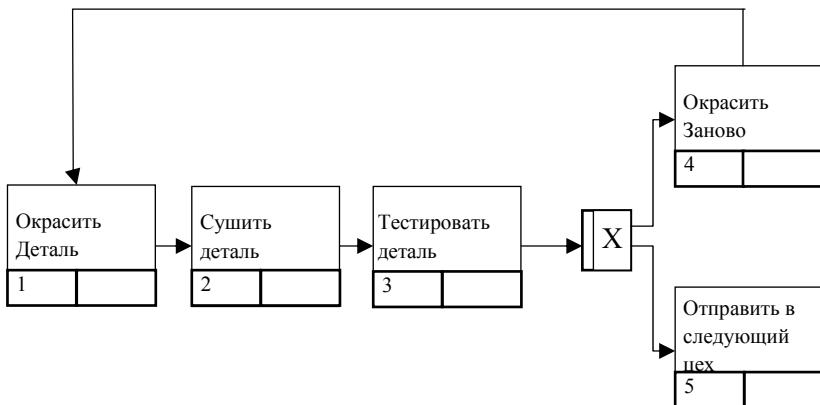


Рис. 3.18. Описание процесса в методологии IDEF3

В отличие от большинства технологий моделирования бизнес-процессов, IDEF3 не имеет жёстких синтаксических или семантических ограничений, делающих неудобным описание неполных или нецелостных систем. Кроме того, автор модели (системный аналитик) избавлен от необходимости смешивать свои собственные предположения о функционировании системы с экспертными утверждениями в целях заполнения пробелов в описании предметной области. На рисунке 3.18 изображён пример описания процесса с использованием методологии IDEF3.

IDEF3 также может быть использован как метод проектирования бизнес-процессов. IDEF3-моделирование органично дополняет традиционное моделирование с использованием стандарта IDEF0. В настоящее время оно получает всё большее распространение как вполне жизнеспособный путь построения моделей проектируемых систем для дальнейшего анализа имитационными методами. Имитационное тестирование часто используют для оценки эксплуатационных качеств разрабатываемой системы. Более подробно методы имитационного анализа будут рассмотрены ниже.

Основой модели IDEF3 служит так называемый сценарий бизнес-процесса, который выделяет последовательность действий или подпроцессов анализируемой системы. Поскольку сценарий определяет назначение и границы модели, довольно важным является подбор подходящего наименования для обозначения действий. Для подбора необходимого имени применяются стандартные рекомендации по предпочтительному использованию глаголов и отглагольных существительных, например «обработать заказ клиента» или «применить новый дизайн».

Сценарий для большинства моделей должен быть документирован. Обычно это название набора должностных обязанностей человека, являющегося источником информации о моделируемом процессе.

Также важным для системного аналитика является понимание цели моделирования, а именно набора вопросов, ответами на которые будет служить модель.

Главной организационной единицей модели IDEF3 является диаграмма. Взаимная организация диаграмм внутри модели IDEF3 особенно важна в случае, когда модель заведомо создаётся для последующего опубликования или рецензирования, что является вполне обычной практикой при проектировании новых систем. В этом случае системный аналитик должен позаботиться о таком информационном наполнении диаграмм, чтобы каждая из них была самодостаточной и в то же время понятной пользователю.

Аналогично другим технологиям моделирования действие, или в терминах IDEF3 «единица работы» (Unit of Work – UOW) – другой важный компонент модели. Диаграммы IDEF3 отображают действие в виде прямоугольника. Как уже отмечалось, действия именуются с использованием глаголов или отглагольных существительных, каждому из действий присваивается уникальный идентификационный номер. Этот номер не используется вновь даже в том случае, если в процессе построения модели действие удаляется. В диаграммах IDEF3 номер действия обычно предваряется номером его родителя (рис. 3.19).

Связи выделяют существенные взаимоотношения между действиями. Все связи в IDEF3 являются однонаправленными, и хотя стрелка может начинаться или заканчиваться на любой стороне блока, обозначающего действие, диаграммы IDEF3 обычно организуются слева направо таким образом, что стрелки начинаются на правой и заканчиваются на левой стороне блоков. В таблице 3.1 приведены три возможных типа связей.

Связь типа «временное предшествование». Как видно из названия, связи этого типа показывают, что исходное действие должно полностью завершиться, прежде чем начнётся выполнение конечного действия. Связь должна быть поименована таким образом, чтобы человеку, просматривающему модель, была понятна причина её появления. Во многих случаях завершение одного действия инициирует начало выполнения другого, как показано на рис. 3.20.

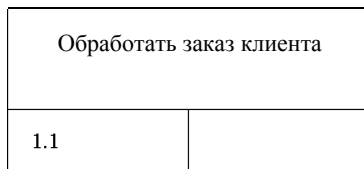


Рис. 3.19. Изображение и нумерация действия в диаграмме IDEF3

3.1. Типы связей

Изображение	Название	Назначение
	Временное предшествование (Temporalprecedence)	Исходное действие должно завершиться, прежде чем конечное действие сможет начаться
	Объектный поток (Objectflow)	Выход исходного действия является входом конечного действия. Из этого, в частности, следует, что исходное действие должно завершиться, прежде чем конечное действие сможет начаться
	Нечёткое отношение (Relationship)	Вид взаимодействия между исходным и конечным действиями задаётся аналитиком отдельно для каждого случая использования такого отношения

В этом примере автор должен принять рекомендации рецензентов, прежде чем начать вносить соответствующие изменения в работу.

Связь типа «объектный поток». Одна из наиболее часто встречающихся причин использования связи типа «объектный поток» заключается в том, что некоторый объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия. Обозначение такой связи отличается от связи временного предшествования двойной стрелкой. Наименования потоковых связей должны чётко идентифицировать объект, который передаётся с их помощью. Временная семантика объектных связей аналогична связям предшествования, это означает, что порождающее объектную связь исходное действие должно завершиться, прежде чем конечное действие может начать выполняться.

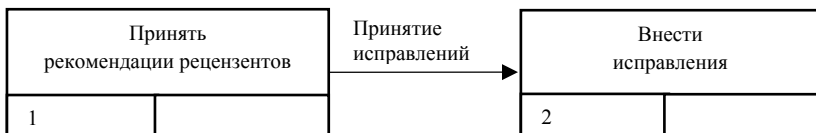


Рис. 3.20. Связь типа «временное предшествование» между действиями 1 и 2

Связь типа «*нечёткое отношение*». Связи этого типа используются для выделения отношений между действиями, которые невозможно описать с использованием предшественных или объектных связей. Значение каждой такой связи должно быть определено, поскольку связи типа «нечёткое отношение» сами по себе не предполагают никаких ограничений. Одно из применений нечётких отношений – отображение взаимоотношений между параллельно выполняющимися действиями. Наиболее часто нечёткие отношения используются для описания специальных случаев связей предшествования, например для описания альтернативных вариантов временного предшествования.




Завершение одного действия может инициировать начало выполнения сразу нескольких других действий или, наоборот, определённое действие может требовать завершения нескольких других действий до начала своего выполнения. Соединения разбивают или соединяют внутренние потоки и используются для описания ветвления процесса:

- разворачивающие соединения используются для разбиения потока. Завершение одного действия вызывает начало выполнения нескольких других;

- сворачивающие соединения объединяют потоки. Завершение одного или нескольких действий вызывает начало выполнения другого действия.

В таблице 3.2 объединены три типа соединений.

3.2. Типы соединений

Графическое обозначение	Название	Вид	Правила инициации
	Соединение «и»	Разворачивающее	Каждое конечное действие обязательно инициируется
		Сворачивающее	Каждое исходное действие обязательно должно завершиться
	Соединение «эксклюзивное «или»	Разворачивающее	Одно и только одно конечное действие инициируется
		Сворачивающее	Одно и только одно исходное действие должно завершиться
	Соединение «или»	Разворачивающее	Одно или несколько конечных действий инициируются
		Сворачивающее	Одно или несколько исходных действий должны завершиться

«И»-соединения. Соединения этого типа инициируют выполнение конечных действий. Все действия, присоединённые к сворачивающему «и»-соединению, должны завершиться, прежде чем начнётся выполнение следующего действия.

Соединение «экслюзивное «или». Вне зависимости от количества действий, связанных со сворачивающим или разворачивающим соединением «экслюзивное «или», инициировано будет только одно из них, и поэтому только оно будет завершено перед тем, как любое действие, следующее за сворачивающим соединением «экслюзивное «или», сможет начаться. Если правила активации соединения известны, они обязательно должны быть документированы либо в его описании, либо пометкой стрелок, исходящих из разворачивающего соединения, как показано на рис. 3.6

Соединение «или» предназначено для описания ситуаций, которые не могут быть описаны двумя предыдущими типами соединений. Аналогично связи нечёткого отношения соединение «или» в основном определяется и описывается непосредственно системным аналитиком.

Указатели – это специальные символы, которые ссылаются на другие разделы описания процесса. Они используются при построении диаграммы для привлечения внимания пользователя к каким-либо важным аспектам модели.

Указатель изображается на диаграмме в виде прямоугольника, похожего на изображение действия. Имя указателя обычно включает его тип (например, ОБЪЕКТ, УОВ и т.п.) и идентификатор (табл. 3.3).

Действия в IDEF3 могут быть декомпозированы или разложены на составляющие для более детального анализа. Метод IDEF3 позволяет декомпозировать действие несколько раз, что обеспечивает документирование альтернативных потоков процесса в одной модели.

Для корректной идентификации действий в модели с множественными декомпозициями схема нумерации действий расширяется и наряду с номерами действия и его родителя включает в себя порядковый номер декомпозиции. Например, в номере действия 1.2.5: 1 – номер родительского действия, 2 – номер декомпозиции, 5 – номер действия.

Рассмотрим построение IDEF3-диаграммы на основании выраженного в текстовом виде описания процесса. Предполагается, что в построении диаграммы принимают участие её автор (в основном как системный аналитик) и один или несколько экспертов предметной области, представляющие описание процесса.

Для экспертов предметной области, подготавливающих описание моделируемого процесса, должны быть документированы границы

3.3. Типы указателей

Тип указателя	Назначение
ОБЪЕКТ (OBJECT)	Для описания того, что в действии принимает участие какой-либо заслуживающий отдельного внимания объект
ССЫЛКА (GOTO)	Для реализации цикличности выполнения действий. Указатель ССЫЛКА может относиться и к соединению
ЕДИНИЦА ДЕЙСТВИЯ (Unit of Behavior – UOB)	Для многократного отображения на диаграмме одного и того же действия. Например, если действие «Подсчёт наличных» выполняется несколько раз, в первый раз оно создаётся как действие, а последующие его появления на диаграмме оформляются указателями UOB
ЗАМЕТКА (NOTE)	Для документирования любой важной информации общего характера, относящейся к изображённому на диаграммах. В этом смысле ССЫЛКА служит альтернативой методу помещения текстовых заметок непосредственно на диаграммах
УТОЧНЕНИЕ (Elaboration – ELAB)	Для уточнения или более подробного описания изображённого на диаграмме. Указатель УТОЧНЕНИЕ обычно используется для описания логики ветвления у соединений

моделирования, чтобы им была понятна необходимая глубина и полнота требуемого от них описания. Кроме того, если точка зрения аналитика на процесс отличается от точки зрения эксперта, это должно быть ясно и подробно обосновано.

Вполне возможно, что эксперты не смогут сделать приемлемое описание без их формального опроса автором модели. В таком случае автор должен заранее подготовить перечень вопросов таким же образом, как журналист для интервью.

Результатом работы экспертов обычно является текстовый документ, описывающий интересующий аналитика круг вопросов. В дополнение к нему может прилагаться письменная документация, позволяющая определить природу изучаемого процесса. Вне зависи-

мости от того, является ли информация текстовой или вербальной, она анализируется и разделяется частями речи для идентификации списка действий (глаголы и отглагольные существительные), составляющих процесс, и объектов (имена существительные), участвующих в процессе.

В некоторых случаях возможно создание графической модели процесса при участии экспертов. Такая модель может быть разработана после сбора всей необходимой информации, что позволяет не отнимать время экспертов на детали форматирования получающихся диаграмм.

Поскольку модели IDEF3 могут одновременно разрабатываться несколькими командами, IDEF3 поддерживает простую схему резервирования номеров действий в модели. Каждому аналитику выделяется уникальный диапазон номеров действий, что обеспечивает их независимость друг от друга.

Если модель создаётся после проведения интервью, аналитик должен принять решение по построению иерархии участвующих в модели диаграмм, например, насколько подробно будет детализироваться каждая отдельно взятая диаграмма. Если последовательность или параллельность выполнения действий окончательно не ясна, эксперты могут быть опрошены вторично (возможно, с использованием черновых вариантов незаконченных диаграмм) для получения недостающей информации. Важно, однако, различать предполагаемую (появляющуюся из-за недостатка информации о связях) и явную (указанную в описании эксперта) неясности.

3.5. КОЛИЧЕСТВЕННОЕ ОПИСАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Шкалы измерения и оценивания. Разработка и эксплуатация информационных, телекоммуникационных и других сложных систем выявили ряд проблем, решить которые можно лишь на основе комплексной оценки различных по своей природе факторов, разнородных связей, внешних условий и т.д.

Поэтому в системном анализе выделяют раздел, который называется теория эффективности (качество систем и процессов). Теория эффективности изучает вопросы количественной оценки качества характеристик и эффективности функционирования сложных систем.

Оценка сложной системы производится для различных целей:

– оптимизация этой системы, т.е. выбор наилучшего алгоритма из нескольких реализующих закон функционирования систем;

- идентификация – определение системы, качество которой наиболее соответствует реальному объекту в заданных условиях;
- для принятия решений по управлению системы.

Оценка – результат, получаемый в ходе некоторого процесса оценивания.

Выделяют четыре этапа оценивания сложных систем:

1. Определение целей оценивания (рассматриваются 2 типа целей: качественные цели, достижение которых выражают в номинальной шкале, и количественные, достижение которых выражают в количественных шкалах). Определение цели должно быть осуществлено относительно системы, для которой рассматривается её элемент.

2. Измерение свойств систем, признанных существенными для целей системы. Выбираются шкалы, и всем исследуемым свойствам присваиваются соответствующие значения на этих шкалах.

3. Обоснование критериев качества и критериев эффективности функционирования системы на основе измеренных на выбранных шкалах свойств.

4. Оценивание – все исследуемые системы рассматриваются как альтернативы, сравниваются по сформулированным критериям и в зависимости от целей оценивания ранжируются, выбираются, оптимизируются и т.д.

Шкала представляет собой множество чисел или символов, с помощью которых можно измерить какую-то отдельную особенность (или свойство) явления, объекта.

Распространёнными являются следующие шкалы:

1. Номинальная шкала – устанавливает взаимно однозначное соответствие между объектами с одним и тем же свойством.

Номинальная шкала основана на отношении эквивалентности и используется для обозначения принадлежности объекта к определённому классу объектов. Номинальная шкала инвариантна относительно однозначных вариантов.

Возможны следующие операции обработки данных, исследованных в номинальной шкале:

1) проверка принадлежности объектов к одному и тому же классу на основе вычислений показателя принадлежности:

$$e_{ij} = \begin{cases} 1, & \text{если } A_i \approx A_j; \\ 0, & \text{если } A_i \not\approx A_j; \end{cases}$$

2) абсолютное и относительное число совпадений объектов, входящих в некоторый класс. $m_k = \sum_{i=1}^m e_{ik}$ – число абсолютных совпадений; $p_k = \frac{m_k}{m}$ – относительное число совпадений;

3) определение номера класса, содержащего наибольшее число объектов.

$$K^{\max} = \arg \max_k p_k.$$

2. Порядковая (ранговая) шкала – устанавливает упорядочивание объектов в зависимости от оцениваемых свойств. Эта шкала используется для обозначения различия объектов по заданным свойствам, на сколько или во сколько один объект превосходит другой, т.е. шкала построена на отношениях строгого порядка, не имеет фиксированного начала отсчёта и масштаба измерений. Эта шкала инвариантна относительно монотонного возрастания (преобразования).

Возможные операции:

- 1) вычисление показателей принадлежности;
- 2) определение целочисленных рангов r_i для объекта:

$$r_i = (m+1) \sum_{j=1}^m r(A_i, A_j);$$

$$r(A_i, A_j) = 1, \text{ если } A_i > A_j, A_i \approx A_j;$$

$$r(A_i, A_j) = 0, \text{ если } A_i < A_j;$$

3) абсолютное и относительное число совпадений;

4) коэффициент ранговой корреляции, характеризует степень близости различных упорядоченных объектов по рангам.

3. Шкала интервалов – устанавливает упорядочивание объектов в зависимости от величины различия какого-либо свойства, при этом шкала интервалов имеет определённый масштаб ($a > 0$) и произвольную точку отсчёта как единицу измерения.

Если 2 объекта имеют оценки j и i свойств объекта, измеряют разницу $d_{ij} = x_i - x_j$.

Инвариантна относительно линейного преобразователя вида $y = \varphi(x) = ax + b$, где a – масштаб; b – начало отсчёта; y – новая шкала.

Возможные операции обработки:

- 1) вычисление показателя принадлежности;

- 2) вычисление относительных и абсолютных совпадений;
- 3) вычисление рангов объектов;
- 4) вычисление коэффициента ранговой корреляции;
- 5) линейные операции над результатами вычислений.

Существуют 2 частных случая данной шкалы:

1) шкала разностей.

$a = 1$; b – любое число.

Пример такой шкалы – время.

Инвариантна относительно $y = \varphi(x) = x + b$;

2) шкала отношений.

Упорядочивание объектов производится по различию свойств (чаще всего «во сколько раз») $a > 0$, $b = 0$.

Сравнение на основании значений и свойств объектов:

$$x_i, x_j \rightarrow h_{ij} = \frac{x_i}{x_j}.$$

Пример такой шкалы – изменение массы.

Инвариантна относительно растяжения: $y = \varphi(x) = ax$, a – любое число.

Возможные операции: те же, что и для шкалы интервалов с добавлением алгебраических операций.

4. Абсолютная шкала. Представляет собой ряд натуральных чисел, по измеренным единицам, упорядочивание в шкале даёт возможность измерить их.

Шкала единственна и инвариантна относительно тождественных преобразований.

Возможные операции: те же, что и для шкалы интервалов.

Помимо данных шкал, инвариантных относительно линейных преобразований, существуют шкалы, инвариантные относительно нелинейных преобразований. Примеры таких шкал: логарифмическая, степенная и т.д.

В зависимости от специфики измеряемых свойств и характеристик объектов, шкалы делятся на количественные (числовые) и качественные (символьные), а также на дискретные и непрерывные.

Тип шкалы	Характер шкалы			
	Количественная	Качественная	Дискретная	Непрерывная
Номинальная		+	+	
Порядковая		+	+	
Интервальная	+		+	+
Отношения	+		+	+

Использование шкал не соответствует специфике измеряемых свойств, сопрягаемых с опасностью получения искажения данных из-за применения при обработке результатов недопустимых операций и сделанных неверных выводов.

Примером ошибочного использования шкал является оцифровка качественной шкалы, т.е. присвоение символьным оценкам номинальной или порядковой шкалы числовых значений.

Обработка оценок, полученных в разных шкалах. Особенностью измерения и оценивания качества сложных систем является то, что для одной и той же системы могут применяться разные типы шкал.

Для получения надёжного измерения показателей проводится несколько измерений одного и того же параметра. Обобщённый показатель качества системы может представлять собой осреднённую величину однородных частных показателей.

При работе с величинами в разных шкалах необходимо соблюдать определённые правила, которые не всегда очевидны, иначе неизбежны неточности, грубые ошибки в измерениях.

Проводить осреднение можно только для однородных характеристик в одной шкале, т.е. усредняются только значения $Y_i, i = \overline{1, n}$, которые представляют собой или оценки различных измерений одной и той же величины или оценки нескольких однородных величин.

Каждое значение Y_i может иметь для исследователя различную ценность, которую учитывают с помощью специального коэффициента значимости $c_i, i = \overline{1, n}$;

$$\sum_{i=1}^n c_i = 1.$$

Тогда для получения осреднённых значений показателя часто используют понятия:

- 1) средневзвешенная арифметическая величина

$$y_{\text{сва}} = \sum_{i=1}^n c_i y_i;$$

- 2) среднеарифметическая величина

$$y_{\text{са}} = \frac{1}{n} \sum_{i=1}^n y_i;$$

3) среднеквадратическое значение

$$y_{\text{СК}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i)^2};$$

4) средневзвешенное геометрическое

$$y_{\text{СВГМ}} = \sqrt[n]{\prod_{i=1}^n y_i^{c_i}};$$

5) среднегеометрическое

$$y_{\text{СГМ}} = \sqrt[n]{\prod_{i=1}^n y_i};$$

6) средневзвешенное гармоническое

$$y_{\text{СВГР}} = \left(\sum_{i=1}^n c_i y_i \right)^{-1};$$

7) среднее гармоническое

$$y_{\text{СГР}} = n \left(\sum_{i=1}^n y_i^{-1} \right)^{-1}.$$

Простые и взвешенные средние величины различают не только по величине, а главное по своей роли в системном анализе.

Средневзвешенные величины используют для сравнения систем с учётом вклада в осреднённую оценку системы.

Среднеарифметические величины используют в случаях, когда важно сравнить абсолютное значение какой-либо характеристики нескольких систем.

Если при замене индивидуальных значений показателя на среднюю величину необходимо сохранить неизменной сумму квадратов величин, то в качестве среднего используется среднеквадратичное (чаще всего используется в статистике).

Среднегеометрическое используется для определения относительной разности отдельных значений при необходимости сокращения

произведения индивидуальной величины. Причём тогда, когда среднее значение качественно одинаково удалено от максимального и минимального значения – когда важен относительный разброс величины.

Среднегармоническое используется, если необходимо, чтобы сумма величин была неизменной. Примером может послужить скорость передачи данных.

Соотношение между разными типами величин определяется по правилу мажоритарности средних величин:

$$СГ_p \leq СГ_m \leq СA \leq СК .$$

Использование необоснованных способов определения средних величин может привести к искусственному занижению или завышению показателя качества системы.

Подходы к оцениванию систем. Ранжирование – процедура упорядочения объектов, выполненная экспертом. На основе своего опыта и знаний эксперт располагает в порядке предпочтения, руководствуясь при этом одним или несколькими показателями сравнения.

Возможны разные варианты упорядочивания объектов.

Ранжирование объектов – измерение некоторых параметров в порядковой шкале. В практике применяется числовое представление последовательности объектов.

Место в последовательности объекта называют рангом объекта.

В зависимости от экспертов определяется общий ранг каждого объекта. Если объекты эквивалентны, то их ранг – среднее арифметическое номеров их позиций по ранговой шкале.

Достоинства ранжирования как метода: простота процедуры, не требующая трудоёмкого обучения экспертов.

Парное сравнение – процедура установления предпочтения объектов при сравнении их попарно. Является простой задачей для эксперта, но более объёмной в целом. При сравнении двух объектов возможно либо бинарное отношение строгого порядка, либо эквивалентность (измерения производятся в порядковой шкале).

Учтём, что порядковая шкала позволяет упорядочить объекты в зависимости от свойств или характеристик объекта. Порядковая шкала даёт возможность установить, что первый объект лучше, хуже или равноценен второму.

В практике используют различные числовые представления для выявления предпочтений, например, матрица парных сравнений (табл. 3.4), построенная на двухуровневой модели ранговой оценки x_{ij} пар объектов a_i, a_j (альтернатив).

3.4. Матрица парных сравнений

$a_i \backslash a_j$	a_1	a_2	a_3	a_4	r_i
a_1	1	2	2	0	5
a_2	0	1	2	0	3
a_3	0	0	1	2	3
a_4	2	2	0	1	5

$$x_{ij} = \begin{cases} 2, & a_i > a_j; \\ 1, & a_i \approx a_j; \\ 0, & a_i < a_j, \end{cases}$$

где x_{ij} – ранговая оценка пары a_i, a_j .

$$r_i = \sum_{j=1}^4 x_{ij} \text{ – ранг объекта, равный сумме ранговых оценок пар в}$$

каждой строке. Тогда $a_1 \approx a_4 > a_2 \approx a_3$.

Если сравнение пар объектов производят по различным свойствам и показателям или производятся сравнения от разных экспертов, тогда по каждому свойству или каждому мнению эксперта составляется своя матрица парных сравнений.

Сравнение всех возможных пар не даёт упорядочивание объектов, возникает задача последующего ранжирования объектов с применением рангов (построение матриц), который может быть суммой показателей по строкам.

На практике эксперт, проводящий парное сравнение, не всегда должен быть последователен (в результате парных сравнений, при определённой предпочтительности объектов, не получается ранжирование и отношение частичного порядка нарушается – не выполняется свойство транзитивности множества). Для устранения этого недостатка используют дополнительный идентификатор. В практике реализуется в рамках метода анализа иерархий, предложенного Саати.

Множественные сравнения. Отличаются от подхода парных сравнений тем, что эксперт последовательно предлагает для сравнения не

пары, а тройки, четвёрки объектов. Эксперт упорядочивает их по значимости или разбивает на классы в целях экспертизы.

Непосредственная оценка. Заключается в присваивании объектам числовых значений в шкале интервалов, а эксперту необходимо поставить в соответствие каждому объекту точку на определённом отрезке числовой оси.

В этом подходе используется шкала отношений, которая позволяет упорядочить объекты не только по различным свойствам, но и сказать, во сколько раз один объект лучше или хуже другого.

Привлечение нескольких экспертов для оценки объекта повлечёт за собой их различные мнения, следовательно, потребуется проверка оценок этих экспертов.

Задания для самопроверки

1. Опишите процедуру Дельфи-метода.
2. Назовите, какие модели берутся за основания декомпозиции.
3. Дайте определение UML.
4. Объясните, из чего исходит функциональная модель.
5. Перечислите типы шкалы.

4. МЕТОДЫ АНАЛИЗА (ОЦЕНКИ) ИНФОРМАЦИОННЫХ СИСТЕМ

4.1. МНОГОКРИТЕРИАЛЬНАЯ ОЦЕНКА СИСТЕМ В УСЛОВИЯХ ОПРЕДЕЛЁННОСТИ

Оценивание систем в условиях определенности производится с использованием методов векторной оптимизации с помощью шкал.

$K = (K_1, K_2, \dots, K_l)$ – векторный критерий.

Отображение множества альтернатив на соответствующих шкалах:

$$K: A \rightarrow R^l,$$

где $K(a)$ – векторная оценка альтернатив; R^l – шкала оценки критериальных показателей альтернатив.

Оценка $K(a)$ альтернатив a по векторному критерию должна быть оптимизирована в области альтернатив A : $K(a) \rightarrow \text{opt} K(a), a \in A$.

Оценка сложных детерминированных систем на основе методов векторной оптимизации проводится в 3 этапа:

1. С помощью методов системного анализа определить частные показатели и критерии эффективности системы.
2. Найти множество оптимальных решений – множество Парето и сформировать задачу оптимизации в общем виде.
3. Провести скаляризацию критериев, т.е. устранить многокритериальность.

Понятие множества Парето. Среди некоего множества альтернатив A можно определить такое A^* , которое включено в множество A , что для его элементов будет выполняться следующее свойство:
 $(\forall a \in A) (\exists a^* \in A^*) (K(a^*) \geq K(a))$.

Смысл этого условия определяет *принцип Парето*.

A^* включает в себя те альтернативы, которые более предпочтительны по сравнению со всеми другими из множества разностей $(A - A^*)$. Любые две альтернативы из A^* по предпочтению несравнимы.

Альтернативы – несравнимы, если альтернатива a_i предпочтительнее a_j по одной группе критериев, а альтернатива a_j лучше a_i по другой группе критериев.

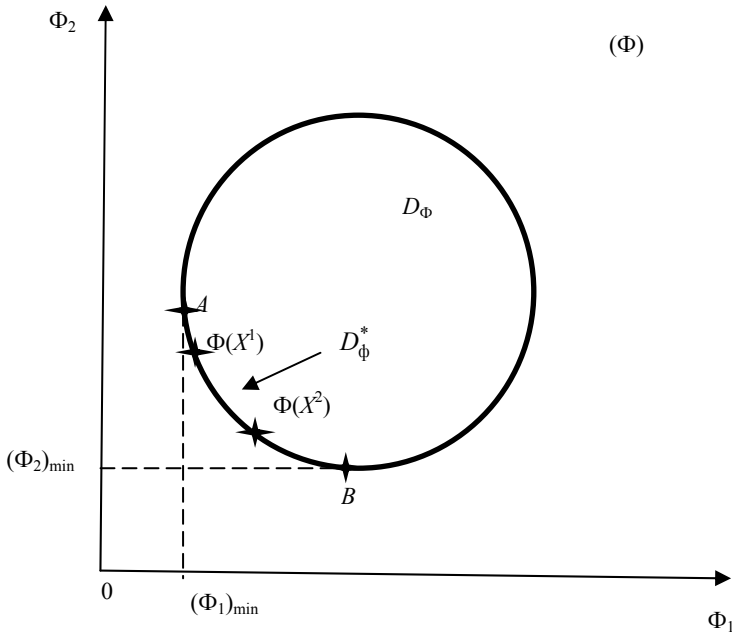


Рис. 4.1. Геометрическая интерпретация множества Парето:

Φ – пространство критериев; D_Φ – область допустимых значений критериев;

D_Φ^* – множество Парето

Исходя из геометрической интерпретации (рис. 4.1), множество Парето можно определить как множество, в котором значение любого из частных критериев оптимальности можно улучшить только за счёт ухудшения других частных критериев – любое из решений, принадлежащее множеству Парето, не может быть улучшено одновременно по всем частным критериям.

Среди $a \in A^*$ невозможно обеспечить минимум по все критериям одновременно, следовательно, необходимо каким-то образом перейти от совокупности критериев к единому их обобщению.

Приёмы такого перехода:

1. Метод выделения главного: ЛПР назначает из всех критериев один – самый главный, остальные выводятся в состав ограничений.

2. Метод лексико-графической оптимизации: критерии, входящие в вектор оптимизации – частные критерии могут быть упорядочены на основе отношения абсолютной предпочтительности для ЛПР.

Пусть критерии пронумерованы так, что № 1 – самый важный критерий, тогда на шаге 1 выбирается первое подмножество $A_1 \in A$, имеющее наилучшие показатели по первому критерию. Если окажется, что количество альтернатив A_1 больше единицы, то следующим шагом выбирается подмножество $A_2 \in A_1$ и рассматривается наилучший показатель по второму критерию. Так продолжается до тех пор, пока не останется самой лучшей. В такой процедуре будут использоваться не все, а лишь наиболее важные критерии, что не всегда может быть оправдано.

3. Метод последовательных уступок: сначала происходит ранжирование критериев по значимости, затем назначается допуск по параметрам и тогда на этапах отбора как в предыдущем методе – из совокупности альтернатив, удовлетворяющих рассмотренному критерию, включают такие, что они входят в допуск по критерию. При этом допуски называют уступками. $\forall j \in N_{j-1}$ число альтернатив $|A_j| > 1$.

Если уступки для всех компонентов критерия равны нулю, то данный метод переходит в метод лексико-графической оптимизации.

4. Человеко-машинные процедуры эффективно используются, учитывая возможности вычислительной техники по быстрому проведению большого объёма расчётов и способность человека воспринимать альтернативы в целом.

В процессе решения задачи выбора альтернатив поиск ведётся во множестве Парето.

Рассмотрим понятие «свертывания» векторного критерия в скалярный.

При этом задача выбора критерия заменяется следующей:

$K(a) \rightarrow \text{extr}_{a \in A} Z$, где $K(a)$ – скалярный критерий, представляет

собой функцию от значений компонентов критерия $K(a) = Z(K_1(a), K_2(a), \dots, K_i(a))$ – операция свёртки.

Основная проблема такого перехода – построение функции свёртки.

Существует четыре особенности задачи свёртки критериев:

1. Обоснование допустимости свёртки:

Учтём, что показатели эффективности функционирования разделяются на 3 группы: результативность; ресурсоёмкость; оперативность.

В общем случае разрешена свёртка показателей, входящих в обобщённый показатель, для каждой группы отдельно. Свёртка показателей из разных групп приводит к потере физического смысла этого критерия.

2. Нормализация критериев: $0 \leq K_i \leq 1$.

3. Учёт приоритетов критериев: задаётся вектор коэффициентов значимости критериев $\gamma = (\gamma_1 \dots \gamma_i)$, $\sum_1^i \gamma = 1$, где γ_i – коэффициент значимости этого критерия. Определение коэффициента значимости сводится либо к использованию формальных процедур, либо к использованию экспертных оценок.

В результате нормализации и учёта приоритетов критериев вместо исходной векторной оценки $K(a)$ образуется новая векторная оценка $k(a)$:

$$K(a) \rightarrow k(a) = Z(\gamma_1 k_1(a), \dots, \gamma_i k_i(a)).$$

Именно эта полученная векторная оценка подлежит преобразованию с использованием функции свёртки. Способ свёртки зависит от характера показателей – аддитивный или мультипликативный.

4. Построение функции свёртки $Z(k_i(a))$ в интересах скалярной оптимизации.

Аддитивная свёртка – представление обобщённого скалярного критерия в виде суммы взвешенных нормированных частных критериев

$$K(a) = \sum_{i=1}^I \gamma_i \frac{k_i(a)}{k_i^0}.$$

Недостатки: не следует из объективной роли частных критериев в определении качества системы. Следовательно, такая свёртка используется как формальный математический приём, придающий задаче удобный вид. В такой свёртке низкие оценки по одному критерию будут компенсироваться высокими оценками по другим критериям, т.е. уменьшение одного критерия до нуля может быть покрыто увеличением другого.

Мультипликативная свёртка – скалярный критерий в виде произведения квадратов частных критериев $K(a) = \prod_{i=1}^I [k_i(a)]^2$.

В мультипликативном критерии схема компромисса предполагает оперирование не с абсолютными, а с относительными изменениями критериев.

Достоинства: не требуется нормировки частных критериев.

Недостатки: имеет место тенденция сглаживать уровни частных критериев за счёт неравнозначных первоначальных критериев.

Выбор между аддитивной и мультипликативной свёртками определяется степенью важности абсолютных или относительных измерений значений частных критериев. Кроме свёртки векторного критерия в теории векторной оптимизации особое место занимает принцип компромисса.

Если из существа задачи оптимизации следует полная недопустимость компенсации значений одних показателей другими, т.е. в задаче требуется обеспечить подтягивание всех показателей к наилучшему уровню, то используется агрегирующая функция следующего вида:

$$K(a) = \text{extr} Z \left(\frac{k_i(a)}{\gamma_i} \right), \gamma_i \neq 0 \text{ для всех } i.$$

Такой показатель используется в задачах планирования, которые называют планированием по узкому месту. При этом в общем случае функция свёртки (агрегированная или усреднённая) может быть представлена как средняя степенная функция:

$$K(a) = \left[\frac{1}{I} \sum_{i=1}^I k_i(a)^P \right]^{1/P}, \text{ где } P \neq 0.$$

P отражает допустимую степень компенсации малых значений одних показателей большими значениями других – чем больше P , тем больше степень возможной компенсации.

Если $P = 1$, то общая свёртка обращается в аддитивную; если $P \rightarrow 0$, то в мультипликативную; если $P \rightarrow \infty$, то в агрегированную; $P \rightarrow 0$ соответствует требованиям обеспечить примерно одинаковый уровень значимости частных критериев.

Вывод. Рассмотренные методы перехода от многокритериальных оценок к скалярным предоставляют широкие возможности для анализа систем. Однако условия применимости тех или иных методов вследствие их эвристического характера не могут быть чётко сформулированы. От этого недостатка свободна другая группа методов, основывающихся на аксиоматическом подходе к принятию решений – методы теории полезности.

Метод анализа иерархий (МАИ) как модель многокритериального выбора. Рациональный выбор альтернатив на основе МАИ предложил Саати. Систематическая процедура состоит в декомпозиции некоторой проблемы на более простые составные части с дальнейшей обработкой последовательных суждений лица, принимающего решения (ЛПР), на основе парных сравнений.

МАИ включает в себя процедуру синтеза множества суждений, получения приоритетности критериев и альтернатив и нахождения рационального варианта решения (рис. 4.2). Для парного сравнения используют шкалу отношений.

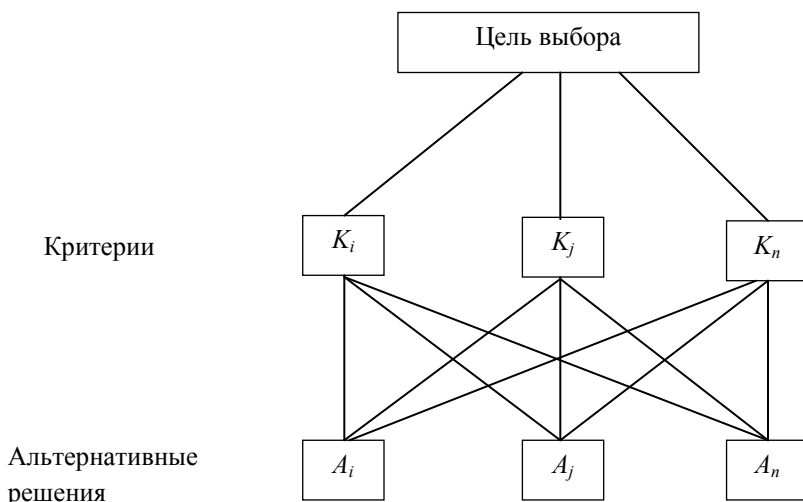


Рис. 4.2. Метод МАИ

Каждая альтернатива оценивается по всем критериям. Этот метод применяется в задачах многокритериального выбора. Реализуется процедура попарного сравнения критериев и альтернатив на основе заданной шкалы отношений.

4.1. Шкала отношений попарных сравнений.

Показатель значимости	Определение показателя	Суть показателя
1	Равная важность	Одинаковая значимость у ЛПР
3	Умеренное превосходство	Небольшое превосходство

Продолжение табл.

Показатель значимости	Определение показателя	Суть показателя
5	Существенное превосходство	Один из элементов явно превосходит другой
7	Значительное превосходство	Один из элементов имеет значительное превосходство над другим
9	Очень сильное превосходство	Безусловное превосходство одного элемента над другим
2, 4, 6, 8	Промежуточные значения превосходства	

4.2. Определение предпочтений по критериям на основе попарных сравнений.

	K_i	K_j	K_n	S_i	X_i	Y_i
K_i	1	ω_{ij}	ω_{in}	S_i	X_i	Y_i
K_j	$1 / \omega_{ij}$	1	ω_{jn}	S_j	X_j	Y_j
K_n	$1 / \omega_{in}$	$1 / \omega_{jn}$	1	S_n	X_n	Y_n
				$\sum_i S_i$		λ_{\max}

После определения суждения ЛПП и заполнения матриц проводится процедура определения вектора приоритетов $[Y_i]$.

1. Собственный вектор матрицы (S_i)

$$S_i = \sqrt[n]{\prod_{i=1}^n \omega_{ij}} ,$$

где $i = \overline{1, n}$, $j = \overline{1, n}$.

2. Нормализованный собственный вектор

$$[X_i] = \frac{S_i}{\sum_i S_i} .$$

3. Вектор приоритетов

$$[Y_i] = [\omega_{ij}]_{n \times n} \times [x_i]_{n \times 1} .$$

4.3. Оценка согласованности суждений.

1. λ_{\max} – наибольшее значение вектора приоритетов

$$\lambda_{\max} = \sum_i y_i.$$

2. ИС – индекс согласованности

$$\text{ИС} = (\lambda_{\max} - n) / (n - 1).$$

3. СС – случайная согласованность, эмпирически определяемый индекс, заданный таблицей.

<i>n</i>	1	2	3	4	5	6	7	8	9	10
СС	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

4. Отношение согласованности $\text{ОС} = \text{ИС}/\text{СС}$, которое должно быть $\text{ОС} \leq (10...20)\%$.

Затем аналогично считаются матрицы попарных сравнений для всех альтернатив, по каждому критерию. Проводится попарное сравнение альтернатив в аспекте заданного критерия. Для каждой матрицы необходимо проверять отношения согласованности суждений. После этого строится матрица глобальных приоритетов.

<i>K</i> \ <i>A</i>	X_i	X_j	X_n	G_i
A_i	X_i^1	X_i^j	X_i^n	G_i
A_j	X_i^1	X_i^j	X_i^n	G_i
A_n	X_i^1	X_i^j	X_i^n	G_i

$$[G_i]_{n \times 1} = [x_i^j]_{n \times n} [x_j]_{n \times 1}.$$

На основе глобальных приоритетов ранжируются альтернативы по предпочтениям ЛПР, т.е. $A_i > A_j$, если $G_i > G_j$.

Тогда из полученной последовательности альтернатив всегда можно определить лучшую.

4.2. ОЦЕНКА СЛОЖНЫХ СИСТЕМ НА ОСНОВЕ ТЕОРИИ ПОЛЕЗНОСТИ

В теории полезности исходят из того, что критерий эффективности предназначен для выявления порядка предпочтений (на исходах операции). Но обеспечить обоснованный выбор решения относительно предпочтения или безразличия непосредственно сравнением альтернатив

затруднительно, так как показатели исходов операции многочисленны, а также имеют различный физический смысл и разные шкалы измерений.

Поскольку на практике не существует универсальной меры с физическим смыслом и позволяющей соизмерить исходы по неравномерной шкале, а потребность существует, то остаётся одно – ввести меру искусственным путём. Такая мера определяется через полезность альтернативы. Своё отношение к полезности альтернативы ЛПР может выразить и количественно, приписав каждому исходу некоторое число, определяющее её относительную предпочтительность.

Таким образом, полезность исхода операции – действительное число, приписываемое альтернативе или исходу операции и характеризующее её предпочтительность по сравнению с другими альтернативами относительно цели.

Зная возможные альтернативы с их показателями полезности, можно построить функцию полезности (ФП), которая даёт основу для сравнения вариантов и выбора решения (рис. 4.3).

Функция полезности (ФП) представляет собой:

$$F(a), A = \{a_k\}, k = \overline{1, l};$$

если $F(a_i) = F(a_j)$, то $a_i \approx a_j$;

если $F(a_i) > F(a_j)$, то $a_i > a_j$.

Каждой альтернативе соответствует свое значение ФП.

Предпочтения ЛПР формируются в виде аксиом, поскольку системы предпочтений могут различаться, разные аксиоматики (наборы аксиом) приводят к различным видам свёртки.

Аксиомы теории полезности:

1. Измеримость – каждой альтернативе может быть поставлено в соответствие неотрицательное действительное число, рассматриваемое как мера относительной полезности этой альтернативы:

$$a_i \rightarrow 0 \leq p_i \leq 1.$$

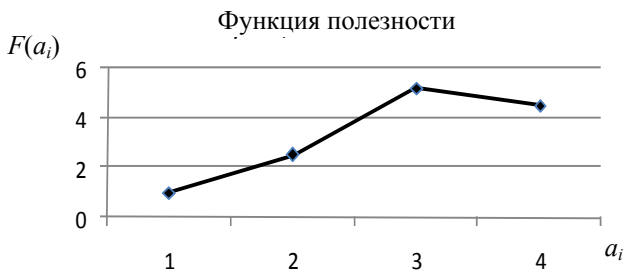


Рис. 4.3. Функция полезности

2. Сравнимость – любые две альтернативы должны быть сравнимы, т.е. для альтернативы всегда возможно определить предпочтительность или эквивалентность

3. Транзитивность – суждения о предпочтениях альтернатив всегда транзитивны, т.е. если $a_i > a_j$ и $a_j > a_k$, то $a_i > a_k$.

4. Коммутативность – предпочтение a_i над a_j не зависит от порядка, в котором они представлены.

5. Независимость – если исход или альтернатива a_i предпочтительнее a_j и кроме того существует исход a_k , который не оценивается относительно a_i и a_j , то

$$a_i(a_k) > a_j(a_k).$$

Согласно теории полезности при выполнении в ходе реальной задачи оценки систем всех пяти аксиом существует такая функция полезности, которая однозначно определена на множестве всех альтернатив с точностью до монотонного строго возрастающего линейного преобразования, т.е. полезность измеряется в шкале интервалов, при этом функция полезности характеризует относительную предпочтительность альтернатив.

В зависимости от типа альтернативы функция полезности может быть непрерывной или дискретной. Функция полезности – прямая, чем больше значения для альтернативы, тем она полезнее. Функция полезности – обратная, чем меньше значения для альтернативы, тем она полезнее.

Все известные способы определения ФП – приближённые и строятся на основе: а) анализа влияния исходов исследуемой операции на операцию более высокого уровня иерархии; б) экспертных оценок; в) аппроксимации.

Анализ влияния исходов на операции более высокого уровня основывается на моделировании и предполагает включение некоторой системы, с помощью которой реализуется исследование операции как элемента в системе на один уровень выше и рассматривается влияние на её функциональность. Такой способ обеспечивает наиболее высокую функциональность.

Использование метода экспертных оценок предполагает, что практический опыт и знания экспертов трудно заменить дедуктивными методами построения формального характера. При любом способе выполнения экспертизы можно выделить 3 этапа:

1. Упорядочивание альтернатив по предпочтительности.
2. Определение полезности каждой альтернативы и проверка на противоречивость.
3. Устранение противоречий в оценках путём корректировок вариантов упорядоченных исходов.

Если ФП определяется с помощью аппроксимации, то при рассмотрении альтернатив отыскиваются точки, соответствующие экстремумам функции полезности, а неизвестные значения между ними заполняются известной зависимостью. Виды аппроксимации выбираются по имеющимся сведениям или качественным соображениям о показателях полезности альтернатив. Наиболее простые виды – одноступенчатая, косинусоидальная, треугольная.

4.2.1. Оценка сложных систем в условиях риска на основе теории полезности

Операции, выполненные в условиях риска, называют вероятностными. Каждой альтернативе ставится в соответствие не один исход, а множество исходов с известными условными вероятностями:

$$a_i \rightarrow \{y_k\} \rightarrow p(y_k/a_i).$$

Эффективность систем в вероятностных операциях находится через математическое ожидание функции полезности на множестве исходов:

$$K(a) = M(F(y)).$$

При совокупности исходов y_k , которые имеют дискретные значения, показывающие эффективность, каждое из которых появляется с условной вероятностью $p(y_k/a_i)$ и имеет функцию полезности $F(y_k)$, выражение для математического ожидания функции полезности имеет вид:

$$K(a_i) = \sum_{k=1}^m p(y_k/a_i) F(y_k).$$

Из этого выражения как частный случай может быть получена оценка эффективности систем для детерминированных операций, если принять, что исход y_k наступит с вероятностью $p = 1$.

На практике удобно задавать условия оценки системы с дискретными величинами в таблице.

a_i	y_k	$p(y_k/a_i)$	$F(y_k)$	$K(a_i)$
a_1	$y_1 \dots y_n$	$p(y_1/a_1) \dots p(y_n/a_1)$	$F(y_1) \dots F(y_n)$	$K(a_1)$
...
a_m	$y_1 \dots y_m$	$p(y_1/a_m) \dots p(y_m/a_m)$	$F(y_1) \dots F(y_m)$	$K(a_m)$

Таким образом, для оценки эффективности систем в условиях риска необходимо:

1. Определить исходы операции по каждой альтернативе y_k .
2. Построить функцию полезности на множестве исходов операций $F(y_k)$.
3. Найти распределение вероятностей исходов на множестве исходов операций $p(y_k/a_i)$.
4. Рассчитать математическое ожидание по формуле

$$K(a_i) = \sum_{k=1}^m p(y_k/a_i) F(y_k).$$

Критерий оптимальности для вероятностных операций – $K(a_i) = \max M(F(y))$.

Процедуру оптимизации по этому критерию называют «оптимизацией в среднем», т.е. оптимизированной системой в условиях риска считается система с максимальным значением математического ожидания функции полезности на множестве всех исходов.

Недостатки: не исключен случай выбора неоптимальной системы для конкретной реализации операции, но если операция будет повторяться многократно, то оптимизированная в среднем система приведёт к наибольшему успеху.

Кроме критерия «в среднем» для оценки вероятностных систем могут использоваться и другие.

4.2.2. Оценка сложных систем в условиях неопределённости

Специфические черты систем не позволяют свести операции, проводимые этими системами к детерминированным, и вынуждают использовать вероятностные характеристики или даже обходиться без них.

К таким чертам относятся:

1. Наличие в управляющей системе в качестве элементов целенаправленных индивидуумов и наличие ЛПП, осуществляющих управление на основе субъективных моделей, что приводит к большому разнообразию поведения системы в целом.

2. Алгоритм управления часто строит сама система управления, преследуя помимо целей, предъявляемых старшей системой, собственные цели.

3. На этапе оценки ситуации в ряде случаев исходят не из фактической ситуации, а из той, модели которой использует ЛПП.

4. В процессе принятия решения существенную роль играют логические рассуждения ЛПП, которые в свою очередь не подчиняются классификации методом формализации.

5. При выборе управляющего воздействия ЛПП может опираться на нечеткие понятия.

6. В большом классе задач управления организационно-техническими системами нет объективных критериев оценивания достижения целей или текущего состояния объекта управления.

Таким образом, несводимость операций к детерминированности не позволяет использовать для их оценки детерминированные критерии.

Общий случай оценки эффективности систем в условиях неопределённости:

a_i	n_j			$K(a_i)$
	n_1	...	n_k	
a_1	k_{11}	k_{12}	k_{1k}	$K(a_1)$
a_2	k_{21}	k_{22}	k_{2k}	$K(a_2)$
...
a_m	k_{m1}	k_{m2}	k_{mk}	$K(a_m)$
p_j	P_1	P_2	P_k	–

n_j – вектор неуправляемых параметров (условия функционирования системы); p_j – вероятность возникновения неуправляемых параметров (условий функционирования); k_{ij} – значение эффективности для i -го варианта системы (альтернативы) в j -х условиях функционирования; $K(a_i)$ – обобщённый критерий эффективности.

Каждая строка таблицы содержит значения эффективности одной системы для всех состояний обстановки, а каждый столбец – значения эффективности всех вариантов системы в заданных (конкретных) условиях функционирования.

При неопределённых операциях могут быть известны множества состояний обстановки и эффективности систем для каждой из них, но могут отсутствовать данные о вероятности появления либо эта вероятность может изменяться.

В зависимости от характера неопределённости операции разделяют на:

1) игровые (неопределённость вносит «как бы» противник своими осознанными действиями) – для исследования таких операций используются методы теории игр;

2) статистически неопределённые (зависят от объективной действительности, например, природа – незаинтересованная сторона) исследуются в теории статистических решений.

Если операция, проводимая системой, уникальна, то для разрешения неопределённости при оценке систем используются субъективные предпочтения ЛПР, поэтому единого критерия оценки для таких неопределённых операций не существует.

Разработаны лишь общие требования к критериям и процедурам оценки и выбора оптимальных систем. Основные требования:

1) оптимальное решение не должно меняться с перестановкой строк и столбцов матрицы эффективности;

2) оптимальное решение не должно меняться при добавлении тождественной строки или столбца;

3) оптимальное решение не должно меняться от добавления постоянного числа к значению каждого элемента матрицы эффективности;

4) оптимальное решение не превращается в неоптимальное и наоборот, в случае добавления новых систем, среди которых нет ни одной более эффективной системы;

5) если две системы оптимальны, то вероятностная смесь этих систем тоже должна быть оптимальной.

В зависимости от характера предпочтений ЛПР наиболее часто в неопределённых операциях используются следующие критерии:

1. Критерий среднего выигрыша.

$$K_{\text{опт}} \rightarrow \max K(a_i), \quad K(a_i) = \frac{1}{k} \sum_{j=1}^k p_j k_{ij}.$$

2. Критерий Лапласа (достаточного основания)

$$K_{\text{опт}} \rightarrow \max K_i(a_i), \quad K(a_i) = \frac{1}{k} \sum_{j=1}^k K_{ij}, j = \overline{1, k}.$$

3. Критерий осторожного наблюдателя (критерий Вальда – «минимакса»)

$$K_{\text{опт}} \rightarrow \min K(a_i), \quad K(a_i) = \max K_{ij}, i = \overline{1, m} \quad j = \overline{1, k}.$$

4. Критерий «максимакса» (оптимизма)

$$K_{\text{опт}} \rightarrow \max (K(a_i)), \quad K(a_i) = \max K_{ij}, i = \overline{1, m} \quad j = \overline{1, k}.$$

5. Критерий «максимина» (пессимизма).

$$K_{\text{опт}} \rightarrow \max (K(a_i)), \quad K(a_i) = \min K_{ij}, i = \overline{1, m} \quad j = \overline{1, k}.$$

6. Критерий пессимизма-оптимизма (критерий Гурвица)

$$K_{\text{опт}} \rightarrow \max (K(a_i)),$$

$$K(a_i) = \alpha \max K_{ij} + (1 - \alpha) \cdot \min K_{ij}, \text{ где } 0 \leq \alpha \leq 1.$$

7. Критерий минимального риска (критерий Севиджа)

$$K_{\text{опт}} = \min_j (\max_i k_{ij}), \quad k_{ij} = \max_i k_{ij} - k_{ij}.$$

8. Критерий Байеса-Лапласа совпадает с критерием среднего выигрыша

$$A^* \in \arg \max_j B(A_j), \quad B(A_i) = \sum_{j=1}^k (p_j k_{ij}).$$

9. Критерий минимума СКО

$$A^* \in \arg \min_j C(A_j), \quad C(A_i) = \left\{ \sum_{j=1}^k \left[k_{ij} - \sum_{t=1}^k p_t k_{it} \right]^2 \right\}^{1/2}.$$

10. Критерий минимума энтропии

$$A^* \in \arg \min_j H(A_j), \quad H(A_i) = - \sum_{j=1}^k p_{ij} \log_2 p_{ij}, \quad p_{ij} = \frac{p_j k_{ij}}{B(A)}.$$

11. Критерий Гермейера

$$A^* \in \arg \max_i \left(\min_j (p_j Z_j(A_i)) \right);$$

$$Z_j(A_i) = a - K_{ij},$$

где $a \geq \max K_{ij}$, если K_{ij} – нормированна, то $a = 1$.

4.3. АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ КОМПЛЕКСНОГО ПРИМЕНЕНИЯ КАЧЕСТВЕННЫХ И КОЛИЧЕСТВЕННЫХ СИСТЕМНЫХ МЕТОДОВ

Методика системного анализа разрабатывается и применяется, если у лица, принимающего решение (ЛПР), нет необходимых сведений об определённой ситуации, позволяющих её формализовать и найти решение задачи.

В современном обществе системные представления уже достигли такого уровня, что мысль о полезности и важности системного подхода к практической деятельности вышла за рамки специальных научных истин и стала привычной, общепринятой. Широко распространилось понимание того, что наши успехи связаны с тем, насколько системно мы подходим к практической деятельности, а наши неудачи вызваны отступлениями от системности.

Системный анализ опирается на системный подход к управлению, а также на ряд математических дисциплин и современных методов управления. В его рамках заключена совокупность определенных практических приёмов решения разнообразных проблем, возникающих во всех сферах целенаправленной деятельности общества. Системный анализ характеризуется упорядоченным, логически обоснованным подходом к исследованию проблем и использованию существующих методов их решения, которые могут быть разработаны в рамках других наук. Он играет роль каркаса, объединяющего все необходимые методы, знания и действия для решения проблем.

В этой ситуации помогают представление объекта в виде системы, привлечение экспертов в различных областях знаний, организация мозговых атак и т.д. Рекомендуется применять различные методы описания систем для создания наиболее эффективного набора методов для данной задачи. Для организации такого процесса определяется часть этапов, выбираются методы для этих этапов, определяются ключевые точки.

SWOT-анализ для определения стратегии развития информационной системы заключается в разделении факторов и явлений на четыре категории: Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы).

Эти факторы проявляются как внутренние и внешние:

Внутренние	Сильные стороны	Слабые стороны
Внешние	Возможности	Угрозы

Но перед тем как разделять факторы и явления на категории, надо понять, зачем мы это делаем. Казалось бы, что тут не ясного? В определении чётко сказано – метод анализа в стратегическом планировании, а, следовательно, цель использования метода – стратегическое планирование. Это не совсем так, или совсем не так. В лучшем случае, правильно проведённый SWOT – это поверхностная инвентаризация тем, которые следовало бы рассмотреть, или вопросов, на которые нужно дать ответ.

Следовательно, проводя SWOT-анализ, нужно понимать, что цель анализа – получение ответа на вопрос, что можно сделать?

Заполнение матрицы анализа можно производить разными путями (например, сначала выписав все факторы и явления отдельно, после разбить их на внутренние и внешние, ну и на последнем шаге уже заносить в таблицу в соответствующую категорию), но есть общие требования: лаконичность описания элемента матрицы; ясность терминов, используемых в описании, для себя и заказчика. Надо внимательно следить, является ли фактор (явление) внутренним или внешним. Часто встречающаяся ошибка – большая клиентская база. Этот фактор можно отнести как к внешним (много покупателей/заказчиков), так и внутренним (обилие клиентов в базе данных, которую нужно обрабатывать). Поэтому надо чётко понимать и явно передавать смысл факторов и явлений.

Ещё надо понимать что возможности – это набор внешних факторов, т.е. если у системы есть возможность не только включать и выключать две лампочки, но и работать с тремя лампочками, это качество самой системы и является внутренним фактором. Приведём пример матрицы SWOT-анализа.

Пример 4.1. SWOT-анализ информационной системы, моделирующей движение транспортных потоков в городе.

Под аппаратной частью (АЧ) будем понимать комплектующие системного блока компьютера, а также монитор.

Под программной частью будем понимать операционную систему.

Внутренние факторы	Сильные стороны (S)	Слабые стороны (W)
	<ol style="list-style-type: none"> 1. Сильная математическая модель, основанная на потоках Пальма. 2. Алгоритм реализован на языке высокого уровня C++. 3. Хорошо структурированный исходный код. 4. Производительная реализация программы под Linux Red Hat. 5. Сверхпроизводительная реализация под строго определённую аппаратную часть.* 6. Использование NFS** 	<ol style="list-style-type: none"> 1. Неочевидный, невыразительный интерфейс. 2. Малое количество поддерживаемых АЧ
Внешние факторы	Возможности (O)	Угрозы (T)
	<ol style="list-style-type: none"> 1. Кроссплатформенность*** (за счёт языка программирования). 2. Увеличение производительности на новых АПЧ.**** 3. Увеличение количества поддерживаемой АЧ 	<ol style="list-style-type: none"> 1. Нет адаптаций под иные ОС. 2. Малая производительность на новых АЧ

* Под строго определённой АЧ понимается заявленный набор моделей комплектующих компьютеров.

** NFS – Network File System – протокол сетевого доступа к файловым системам.

*** Возможность переноса на другие платформы.

**** Новые АПЧ – АЧ, не входящие в строго определённый набор моделей комплектующих компьютеров, но поддерживаемые ИС и ПЧ, отличающиеся принципами работы от ОС Linux Red Hat.

Далее необходимо проанализировать матрицу, сопоставив каждый элемент Сильных сторон (S), Слабых сторон (W), Возможностей (O) и Угроз (T) и на основе сопоставления получить рекомендации.

В нашем примере попарный анализ даёт следующие рекомендации:

Факторы O–T:

1) O1–T1. Реализовать кроссплатформенность языка программирования с целью создания адаптаций под иные ОС. (Появится угроза малой производительности на новых ОС T3).

2) O2–T2. Увеличить производительность АПЧ, снизив угрозы малой производительности АПЧ (T2).

Факторы S–W:

1) S3–W2. Хорошо структурированный исходный код позволит быстро наращивать количество поддерживаемых АЧ. (Увеличится угроза малой производительности АЧ T2).

2) S1–W1. Расширение математической модели с целью устранения неочевидности интерфейса. Например, добавление косвенных параметров, которые имеют более понятную природу или легче вычислимы (например, вместо интенсивности потоков на различных маршрутах будет браться количество владельцев авто).

3) S3–W1. Хорошо структурированный исходный код позволит изменить интерфейс (O4).

Факторы S–T:

1) S5–T2. Увеличение количества сверхпроизводительной реализации под строго определённую аппаратную часть снизит угрозы низкой производительности АЧ (T2).

2) S6–T2. Детальная настройка NFS снизит важность угрозы низкой производительности АПЧ (T2).

Факторы O–W:

1) O4–W1. Изменить интерфейс с целью повышения его «дружелюбности» (уменьшение неочевидности, невыразительности W1).

2) O3–W2. Выбрать новые АПЧ (снизив недостаток W2, повысим угрозу T2).

Рекомендации:

1) SO.01: S.02, S.05, S.03 & O.01 – учитывая расположение интернет-кафе вблизи института, невысокие цены, доступ к высокоскоростному интернету, заключить договор с институтом на проведение факультативов.

2) SO.02: S.04&O.02 – в связи с наличием квалифицированного персонала создать обучающие курсы по работе с компьютером.

3) OW.01: O.01, O.02&W.02, W.03 – за счёт вырученных денег с обучающихся курсов и проведённых факультативов погасить кредит и оплатить аренду.

4) OW.02: O.02&W.04 – за счёт полученных денег с проведённых курсов создать свой web-сайт.

5) WT.01: W.04&T.02 – за счёт создания своего сайта производить рекламу своих услуг обучающих курсов по работе с компьютером.
 Пример 4.2. Анализ интернет-клуба.

Сильные стороны (S)	Слабые стороны (W)
S.01: Стабильный доход. S.02: Расположение около института. S.03: Наличие скоростного доступа к интернету. S.04: Наличие квалифицированного персонала. S.05: Невысокие цены. S.06: Наличие наиболее популярных компьютерных игр	W.01: При разрыве кабеля или других неполадках в соединении вся сеть прекращает работу. W.02: Высокая стоимость аренды. W.03: Высокий процент по кредиту, привлечённому для покупки оборудования. W.04: Отсутствие своего Web-сайта
Возможности (O)	Угрозы (T)
O.01: Заключение договора с институтом на проведение факультативов. O.02: Создание обучающих курсов по эксплуатации компьютера. O.03: Проведение ряда скидок	T.01: Быстрый рост конкурентов. T.02: Снижение стоимости домашнего интернет, вследствие чего идёт падение спроса на услуги, предоставляемые интернет-клубом

Далее к полученным рекомендациям применяется метод морфологического анализа путём декомпозиции каждой рекомендации и составления морфологического ящика с вариантами (альтернативами) реализации на каждом уровне декомпозиции.

Для анализа и рационального выбора вариантов реализации рекомендаций целесообразно использовать метод анализа иерархий (метод Саати).

Эти два метода позволят ответить на вопрос, как именно реализовать рекомендации стратегического развития системы, т.е. в полном объёме достичь цели системного анализа.

4.4. МЕТОДЫ СТАТИСТИЧЕСКОЙ ОЦЕНКИ ИНФОРМАЦИОННЫХ СИСТЕМ

Интервальное оценивание – способ получения оценки для неизвестного значения скалярного параметра с помощью интервала его допустимых значений и определения вероятности того, что в этом ин-

тервале находится истинное значение параметра. На практике для получения интервальной оценки параметра q обычно заранее выбирается число p , такое, что $0 < p < 1$, и находятся два других числа, зависящих от результатов наблюдений $q_1(q^*)$ и $q_2(q^*)$, таких, что вероятность нахождения q в интервале (q_1, q_2) равна p : $P[q_1(q^*) \leq q \leq q_2(q^*)] = p$. В этом случае интервал (q_1, q_2) называется $(100 \cdot p)$ -процентным доверительным интервалом.

Вероятность того, что доверительный интервал содержит истинное значение параметра q , равная p , называется *коэффициентом доверия*. Величины $q_1(q^*)$ и $q_2(q^*)$ называются соответственно нижней и верхней доверительными границами для параметра q . Интервальное оценивание применяется как альтернатива точечному оцениванию параметра ошибки, т.е. доверительный интервал для q соответствует ошибке параметра q . Ширина интервала (разность между двумя границами) является мерой точности оценки.

Существует большой набор процедур доверительного оценивания, которыми можно заменить традиционные критерии проверки гипотез, используемых в классических ситуациях: доверительный интервал для математического ожидания, дисперсии, вероятности, коэффициенты множественной корреляции (детерминации) при оценке силы связи между переменными и т.д.

Проверка статистических гипотез. Статистическая гипотеза представляет собой некоторое предположение о законе распределения случайной величины или о параметрах этого закона, формулируемое на основе выборки. Примерами статистических гипотез являются предположения: генеральная совокупность распределена по экспоненциальному закону; математические ожидания двух экспоненциально распределённых выборок равны друг другу.

В первой из них высказано предположение о виде закона распределения, а во второй – о параметрах двух распределений. Гипотезы, в основе которых нет никаких допущений о конкретном виде закона распределения, называют *непараметрическими*, в противном случае – *параметрическими*.

Гипотезу, утверждающую, что различие между сравниваемыми характеристиками отсутствует, а наблюдаемые отклонения объясняются лишь случайными колебаниями в выборках, на основании которых производится сравнение, называют *нулевой* (основной) гипотезой и обозначают H_0 . Наряду с основной гипотезой рассматривают и *альтернативную* (конкурирующую, противоречащую) ей гипотезу H_1 . Если нулевая гипотеза будет отвергнута, то будет иметь место альтернативная гипотеза.

Различают простые и сложные гипотезы. Гипотезу называют *простой*, если она однозначно характеризует параметр распределения случайной величины. *Сложной* называют гипотезу, которая состоит из конечного или бесконечного множества простых гипотез.

Проверка гипотезы основывается на вычислении некоторой случайной величины – критерия, точное или приближённое распределение которого известно. Обозначим эту величину через z , её значение является функцией от элементов выборки $z = z(x_1, x_2, \dots, x_n)$. Процедура проверки гипотезы предписывает каждому значению критерия одно из двух решений – принять или отвергнуть гипотезу. Тем самым всё выборочное пространство и соответственно множество значений критерия делятся на два непересекающихся подмножества S_0 и S_1 . Если значение критерия z попадает в область S_0 , то гипотеза принимается, а если в область S_1 , то гипотеза отклоняется. Множество S_0 называется *областью принятия гипотезы или областью допустимых значений*, а множество S_1 – *областью отклонения гипотезы или критической областью*. Выбор одной области однозначно определяет и другую область.

Принятие или отклонение гипотезы H_0 по случайной выборке соответствует истине с некоторой вероятностью и, соответственно, возможны два рода ошибок. *Ошибка первого рода* возникает с вероятностью a тогда, когда отвергается верная гипотеза H_0 и принимается конкурирующая гипотеза H_1 . *Ошибка второго рода* возникает с вероятностью b в том случае, когда принимается неверная гипотеза H_0 , в то время как справедлива конкурирующая гипотеза H_1 . *Доверительная вероятность* – это вероятность не совершить ошибку первого рода и принять верную гипотезу H_0 . Вероятность отвергнуть ложную гипотезу H_0 называется *мощностью критерия*. Следовательно, при проверке гипотезы возможны четыре варианта исходов.

Варианты исходов при проверке гипотез

Гипотеза H_0	Решение	Вероятность	Примечание
Верна	Принимается	$1 - a$	Доверительная вероятность
	Отвергается	a	Вероятность ошибки первого рода
Неверна	Принимается	b	Вероятность ошибки второго рода
	Отвергается	$1 - b$	Мощность критерия

Основы планирования эксперимента. Критерии адекватности модели объекту предметной области. Планирование эксперимента (англ. *Experimental design techniques*) – комплекс мероприятий, направленных на эффективную постановку опытов. *Основная цель* планирования эксперимента – достижение максимальной точности измерений при минимальном количестве проведённых опытов и сохранении статистической достоверности результатов. Планирование эксперимента применяется при поиске оптимальных условий, построении интерполяционных формул, выборе значимых факторов, оценке и уточнении констант теоретических моделей и др.

Принципы, положенные в основу теории планирования эксперимента, направлены на повышение эффективности экспериментирования, т.е. стремление к минимизации общего числа опытов; одновременное варьирование всеми переменными, определяющими процесс, по специальным правилам – алгоритмам; использование математического аппарата, формализующего многие действия экспериментатора; выбор чёткой стратегии, позволяющей принимать обоснованное решение после каждой серии экспериментов.

Для проведения эксперимента любого типа необходимо: разработать гипотезу, подлежащую проверке; создать программы экспериментальных работ; определить способы и приёмы вмешательства в объект исследования; обеспечить условия для осуществления процедуры экспериментальных работ; разработать пути и приёмы фиксирования хода и результатов эксперимента (приборы, установки, модели и т.п.); обеспечить эксперимент необходимым обслуживающим персоналом.

Первоначальный анализ данных начинается с попытки описания свойств изучаемой характеристики в наиболее компактном и информативном виде. Это основа для дальнейшего аналитического исследования. Для его проведения исходные данные представляют в виде дискретного или интервального *вариационного ряда* (статистического ряда распределения) – упорядоченной последовательности измеренных значений $x_1 < x_2 < \dots < x_n$ и частот их встречаемости m_i . В случае большого количества различных значений или когда измеряемая характеристика принимает непрерывный ряд значений строят *интервальный ряд* распределения.

Методами *описательной статистики* принято называть методы описания выборок с помощью различных показателей и графиков.

1. *Показатели положения* описывают положение данных на числовой оси. Примеры таких показателей – минимальный и максимальный элементы выборки (первый и последний член вариационного ряда), верхняя и нижняя квантили (они ограничивают зону, в которую

попадают 50% центральных элементов выборки). Наконец, сведения о середине совокупности могут дать выборочное среднее значение, выборочную медиану и другие аналогичные характеристики.

2. *Показатели разброса* описывают степень разброса данных относительно своего центра. К ним, в первую очередь, относятся: дисперсия выборки, стандартное отклонение, размах выборки (разность между максимальным и минимальным элементами), межквантильный размах (разность между верхней и нижней квантилью), коэффициент эксцесса и т.п. По сути дела, эти показатели говорят, насколько основная масса данных группируется около центра.

3. *Показатели асимметрии*. Третья группа показателей отвечает на вопрос о симметрии распределения данных около своего центра. К ней можно отнести: коэффициент асимметрии, положение выборочной медианы относительно выборочного среднего и относительно выборочных квантилей, гистограмму и т.д.

4. *Показатели, описывающие закон распределения*. Наконец, четвёртая группа показателей описательной статистики даёт представление собственно о законе распределения данных. Сюда относятся графики гистограммы и эмпирической функции распределения, таблицы частот.

При обработке результатов *прямых измерений* используется следующий порядок операций:

1. Результат каждого измерения записывается в таблицу.
2. Вычисляется среднее значение из n измерений:

$$\bar{x} = \sum x_i / n .$$

3. Находится погрешность отдельного измерения:

$$\Delta x_i = \bar{x} - x_i .$$

4. Вычисляются квадраты погрешностей отдельных измерений: $(\Delta x_1)^2, (\Delta x_2)^2, \dots, (\Delta x_n)^2$.

5. Определяется среднеквадратичная ошибка среднего арифметического:

$$\bar{x}' = \sqrt{\frac{\sum (\Delta x_i)^2}{n(n-1)}} .$$

6. Задаётся значение надёжности (обычно берут $P = 0,95$).
7. Определяется табличный коэффициент Стьюдента t для заданной надёжности P и числа произведённых измерений n .
8. Находится доверительный интервал (погрешность измерения):

$$\Delta x = \bar{x}' t .$$

9. Если величина погрешности результата измерения Δx оказывается сравнимой с величиной погрешности прибора δ , то в качестве границы доверительного интервала берётся:

$$\Delta x = \sqrt{(S_{\bar{x}}t)^2 + \delta^2}.$$

10. Если одна из ошибок меньше другой в три или более раз, то меньшую отбрасывают. Окончательный результат записывается в виде:

$$x = \bar{x}' \pm \Delta x.$$

11. Оценивается относительная погрешность результата измерений:

$$\varepsilon = \frac{\Delta x}{x} 100\%.$$

При обработке результатов *косвенных измерений* применяется следующий порядок операций:

1. Все величины, находимые прямыми измерениями, обрабатываются в соответствии с правилами обработки результатов прямых измерений. При этом для всех измеряемых величин задаётся одно и то же значение надёжности P .

2. Оценивается точность результата косвенных измерений по формулам, где производные вычислены при средних значениях величин (*систематическая ошибка*):

$$\delta N = \pm \left[\left| \frac{\partial f}{\partial x} \partial x \right| + \left| \frac{\partial f}{\partial y} \partial y \right| + \left| \frac{\partial f}{\partial z} \partial z \right| + \dots \right]$$

или

$$\delta N = \pm \left[\left| \frac{\partial \ln f}{\partial x} \partial x \right| + \left| \frac{\partial \ln f}{\partial y} \partial y \right| + \left| \frac{\partial \ln f}{\partial z} \partial z \right| + \dots \right],$$

где $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial z}$, ... – частные производные функции одной или не-

скольких непосредственно измеряемых величин $N = f(x, y, z, \dots)$ по аргументу x, y, z, \dots , найденные в предположении, что все остальные аргументы, кроме того, по которому находится производная, постоянные; $\delta x, \delta y, \delta z$ – систематические ошибки аргументов.

Если ошибка отдельных измерений входит в результат дифференцирования несколько раз, то надо сгруппировать все члены, содержащие одинаковый дифференциал, и выражения в скобках, стоящие перед дифференциалом, взять по модулю; знак d заменить на Δ (или δ).

Первой формулой удобно пользоваться в случае, если функция имеет вид суммы или разности аргументов. Вторую формулу применять целесообразно, если функция имеет вид произведения или частного аргументов.

Для нахождения *случайной ошибки* косвенных измерений следует пользоваться формулами:

$$\Delta N = \pm \sqrt{\left(\frac{\partial f}{\partial x} \Delta x\right)^2 + \left(\frac{\partial f}{\partial y} \Delta y\right)^2 + \left(\frac{\partial f}{\partial z} \Delta z\right)^2 + \dots}$$

или

$$\Delta N = \pm \bar{N} \sqrt{\left(\frac{\partial \ln f}{\partial x} \Delta x\right)^2 + \left(\frac{\partial \ln f}{\partial y} \Delta y\right)^2 + \left(\frac{\partial \ln f}{\partial z} \Delta z\right)^2 + \dots},$$

где Δx , Δy , Δz , ... – доверительные интервалы при заданных доверительных вероятностях (надёжностях) для аргументов x , y , z , Следует иметь в виду, что доверительные интервалы Δx , Δy , Δz , ... должны быть взяты при одинаковой доверительной вероятности $P_1 = P_2 = \dots = P_n = P$. В этом случае надёжность для доверительного интервала Δ_N будет тоже P .

Первой формулой удобно пользоваться в случае, если функция $N = f(x, y, z, \dots)$ имеет вид суммы или разности аргументов. Вторую формулу применять целесообразно, если функция $N = f(x, y, z, \dots)$ имеет вид произведения или частного аргументов.

Часто наблюдается случай, когда систематическая ошибка и случайная ошибка близки друг к другу, и они обе в одинаковой степени определяют точность результата. В этом случае общая ошибка S находится как квадратичная сумма случайной Δ и систематической δ ошибок с вероятностью не менее чем P , где P – доверительная вероятность случайной ошибки:

$$S = \sqrt{\Delta^2 + \delta^2}.$$

3. Если случайная и систематическая ошибки по величине близки друг к другу, то они складываются по правилу сложения ошибок. Если одна из ошибок меньше другой в три или более раз, то меньшую отбрасывают. Окончательный результат записывается в виде:

$$N = f(x, y, z, \dots) \pm \Delta f.$$

4. Оценивается относительная погрешность результата серии косвенных измерений: $e = (\Delta f / f) \cdot 100\%$.

Интеллектуализация процесса анализа выборочных данных в эксперименте. Большинство методов интеллектуального анализа данных было разработано в рамках теории искусственного интеллекта в 70–80-х годах, но получили распространение только в последние годы, когда проблема интеллектуализации обработки больших и быстро растущих объемов корпоративных данных потребовала их использования в качестве надстройки над хранилищами данных.

В основу современной технологии *Data Mining (discovery-driven data mining)* положена концепция шаблонов, отражающих фрагменты многоаспектных взаимоотношений в данных. Эти шаблоны представляют собой закономерности, свойственные выборкам данных, которые могут быть компактно выражены в понятной человеку форме. Поиск шаблонов производится методами, не ограниченными рамками априорных предположений о структуре выборки и виде распределений значений анализируемых показателей.

Важное положение Data Mining – нетривиальность разыскиваемых шаблонов. Это означает, что найденные шаблоны должны отражать неочевидные, неожиданные регулярности в данных, составляющие так называемые скрытые знания. Data Mining – это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности (рис. 4.4).

Данная технология применяется конечными пользователями и аналитиками. Конечные пользователи используют системы Data Mining для получения информационных электронных таблиц. Аналитики знают, как интерпретировать данные и сделать необходимые вычисления для анализа информации. Данный класс пользователей выполняет три вида анализа: периодический, стратегический и непрерывный.

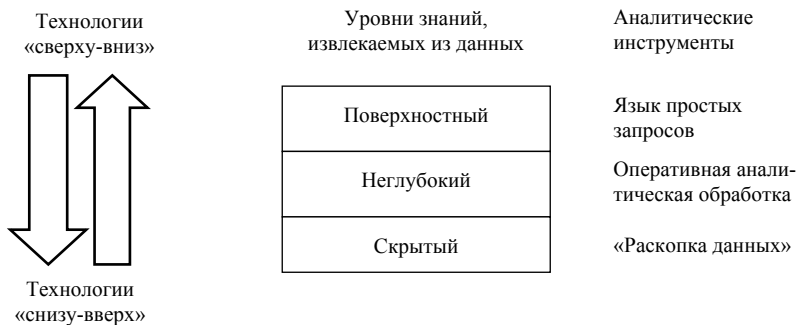


Рис. 4.4. Уровни знаний, извлекаемых из данных

Периодический анализ используется для определения зарождающихся тенденций развития. Данные для анализа берутся из определённого периода. *Стратегический* анализ делается на больших объёмах данных для понимания специфики происхождения тех или иных событий. *Непрерывный* анализ используется для получения факторов, которые оказывают воздействие на течение деловых процессов.

Интеллектуальный анализ данных можно разделить на три стадии: выявление закономерностей; использование выявленных закономерностей для предсказания неизвестных значений; анализ исключений (рис. 4.5).

Существуют два типа статистического анализа: исследовательский и подтверждающий. В *подтверждающем* анализе имеется гипотеза, и он подтверждает или отвергает её. *Исследовательский* анализ отыскивает подходящие гипотезы для подтверждения или опровержения, при этом аналитическая система берёт инициативу на себя при анализе данных, без участия в этом процессе пользователя (например, OLAP (*Online Analytical Processing*) система).

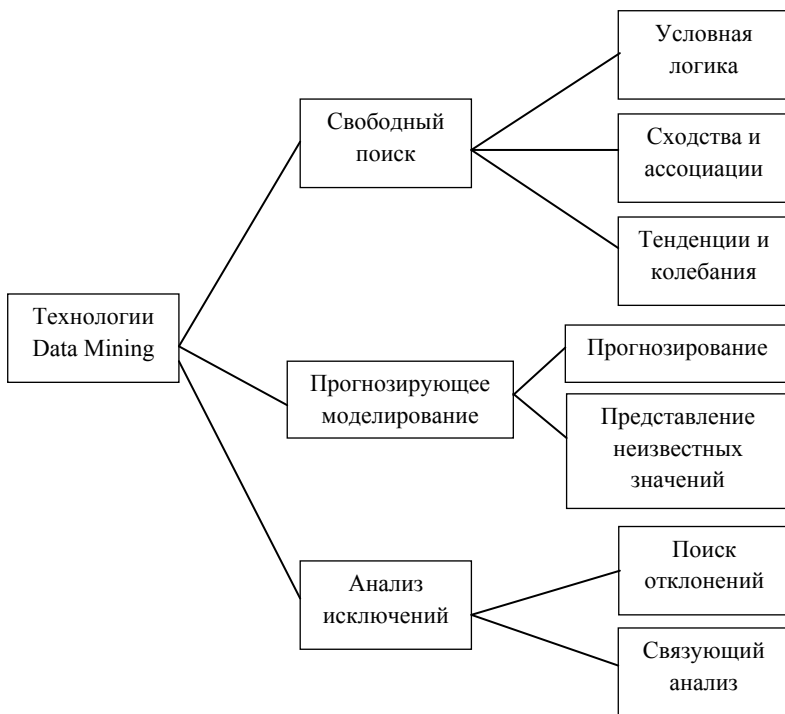


Рис. 4.5. Составляющие стадии технологии Data Mining



Рис. 4.6. Подходы технологии Data Mining

Задачи интеллектуального анализа данных можно разделить по типу извлекаемой информации: классификация; кластеризация; выявление ассоциаций; выявление последовательностей; прогнозирование (рис. 4.6).

Основные методики, используемые при интеллектуальном анализе данных, могут быть выделены в отдельные группы, каждая из которых имеет свои преимущества и области применения. Современные методики строятся как на выделении шаблонов, так и на сохранении и длительном использовании данных. Выделение шаблонов может быть разделено на три группы методик: логическая, вычислительная и перекрестные таблицы. Гибридные или смешанные подходы считают наилучшими, соединяя в себе логику и уравнения с многомерным анализом.

Задачи для самопроверки

1. Дайте определение множеству Парето.
2. Сформулируйте аксиомы теории полезности.
3. Поясните, в чём заключается SWOT-анализ.
4. Объясните, чем является статистическая гипотеза.

5. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СИСТЕМНОГО АНАЛИЗА И МОДЕЛИРОВАНИЯ

5.1. СРЕДА РАЗРАБОТКИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МОДЕЛЕЙ UMBRELLO UML MODELLER

Для выполнения лабораторных работ по объектно-ориентированному моделированию предлагается использовать среду Umbrello, которая удовлетворяет двум ключевым требованиям: бесплатность и кроссплатформенность.

Umbrello – среда UML-моделирования. Это приложение является свободным программным обеспечением, предназначенным для построения UML-диаграмм на платформе Unix. Является CASE-инструментом. Umbrello входит в пакет kdesdk оконного менеджера KDE, но также хорошо работает и с другими оконными менеджерами. Пользовательский интерфейс программы прост и функционален.

Umbrello поддерживает все стандартные типы UML-диаграмм. Также поддерживается импорт из C++, IDL, Pascal/Delphi, Ada, Python, Java, Perl (с помощью внешнего инструмента) и экспорт диаграмм в различные языки программирования. Формат файла, используемый при хранении диаграмм, основан на XML.

Umbrello позволяет хранить данные модели в форматах DocBook и XHTML. Это было сделано с целью поддержки модели совместной разработки, когда не все разработчики имеют доступ к Umbrello. Эта возможность незаменима при необходимости размещения контента модели на web-сайте.

Окно программы представлено на рис. 5.1 и имеет следующие основные элементы управления:

1. Меню создания (добавления) UML-диаграмм в проект.
2. Меню выбора языковых примитивов открытой диаграммы.
3. Закладки открытых диаграмм проекта.
4. Иерархическое дерево типов диаграмм, созданных в проекте.
5. Рабочая область (Полотно языкового выражения).

Диаграмма классов является основным логическим представлением модели и содержит детальную информацию о внутреннем устройстве объектно-ориентированной программной системы или, используя современную терминологию, об архитектуре программной системы. Активизировать рабочее окно диаграммы классов можно несколькими способами:

– окно диаграммы классов появляется по умолчанию в рабочем окне диаграммы после создания нового проекта;

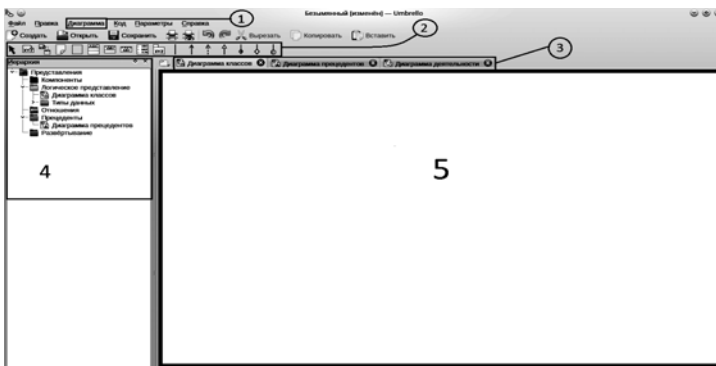






Рис. 5.1. Главное окно среды UML-моделирования Umbrello

- раскрыть логическое представление (Logical View) в браузере проекта и выделить элемент class diagram;
- выбрать пункт главного меню Diagram->New->class diagram (Диаграмма->Новая->Диаграмма классов).

При этом появляется новое окно с чистым рабочим листом диаграммы классов и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы классов. Назначение отдельных кнопок панели можно узнать также из всплывающих подсказок. (табл. 5.1).

5.1. Назначение кнопок специальной панели инструментов диаграммы классов

Графическое изображение	Всплывающая подсказка	Назначение кнопки
	Select	Превращает изображение курсора в стрелку для последующего выделения элементов на диаграмме
	Note	Добавляет на диаграмму примечание
	Anchor	Добавляет связь примечания с соответствующим графическим элементом диаграммы
	Label	Добавляет на диаграмму текстовую область

Графическое изображение	Всплывающая подсказка	Назначение кнопки
	Box	Добавляет на диаграмму коробку (пакет)
	Class	Добавляет на диаграмму класс
	Interface	Добавляет на диаграмму интерфейс
	Datatype	Добавляет на диаграмму тип данных
	Enum	Добавляет на диаграмму перечисление
	Package	Добавляет на диаграмму пакет
	Association	Добавляет на диаграмму ассоциацию
	Directional Association	Добавляет на диаграмму направленную ассоциацию
	Dependency	Добавляет на диаграмму отношение зависимости
	Implements	Добавляет на диаграмму отношение принадлежности
	Composition	Добавляет на диаграмму отношение композиции
	Aggregation	Добавляет на диаграмму отношение агрегации
	Containment	Добавляет на диаграмму отношение «содержится»

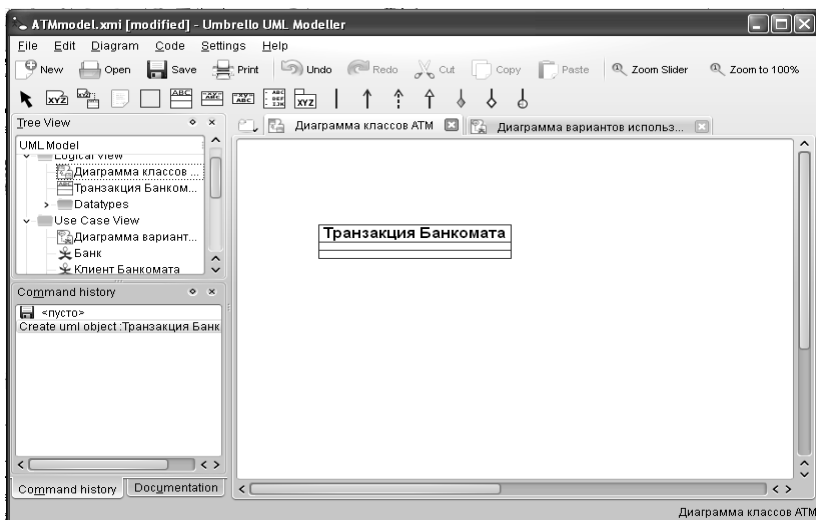


Рис. 5.2. Диаграмма классов модели банкомата после добавления на неё класса «Транзакция Банкомата»

Рассмотрим добавление класса на диаграмму классов и редактирование его свойств.

Для добавления класса на диаграмму классов нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы класса на специальной панели инструментов, отпустить левую кнопку мыши и щёлкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение класса с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию NewClass. Продолжая разработку модели банкомата в качестве сквозного примера проекта, построим для этой модели следующую каноническую диаграмму – диаграмму классов. С этой целью следует изменить предложенное по умолчанию имя диаграммы «class diagram» на «Диаграмма классов ATM», а имя добавленного на диаграмму класса – на «Транзакция Банкомата».

Поскольку разрабатываемая модель банкомата на начальных этапах работы над проектом используется для анализа общей архитектуры проекта и согласования ее с различными участниками рабочей группы, имена классов, их атрибутов и операций для большей наглядности и понимания задаются на русском языке с пробелами и записываются символами кириллицы. В последующем по мере выполнения проекта и реализации модели на некотором языке программирования,

имена соответствующих классов, атрибутов и операций должны быть преобразованы в символы латиницы. При этом имена этих элементов модели должны быть записаны без пробелов. В контексте управляемой моделью архитектуры первую модель еще называют независимой от платформы реализации, а вторую – зависимой от платформы реализации. Для класса «Транзакция Банкомата» можно уточнить его назначение в модели с помощью указания стереотипа и пояснительного текста в форме документации. С этой целью следует щелчком правой кнопки мыши на изображении этого класса на диаграмме открыть диалоговое окно спецификации свойств этого класса (рис. 5.3) и на вкладке General (Общие) указать в категории «Stereotype name» стереотип entity (сущность).

Выбор данного стереотипа означает, что соответствующий класс предназначен для хранения информации, которая должна сохраняться в системе после уничтожения объектов данного класса. Далее в секцию документации данного класса можно ввести поясняющий текст: «Используется для сохранения информации о выполненных банкоматом транзакциях» и нажать кнопку Apply или OK, чтобы сохранить результаты редактирования свойств выбранного класса. После назна-

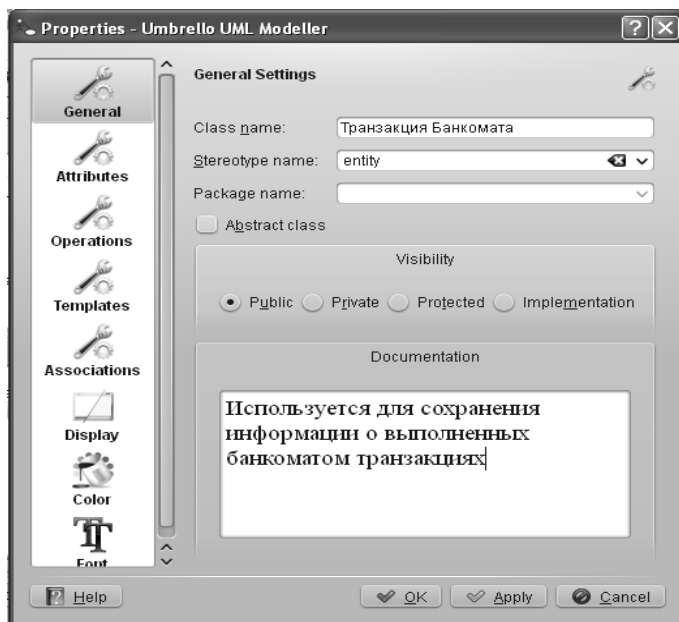


Рис. 5.3. Диалоговое окно свойств класса

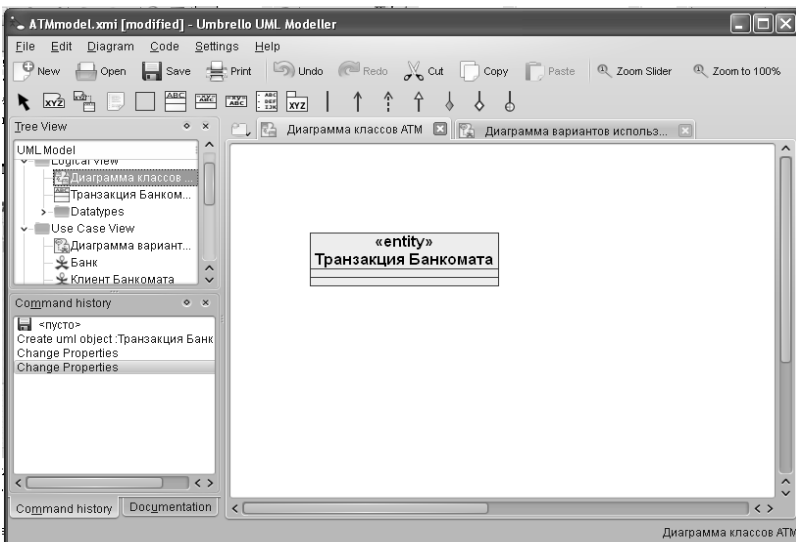


Рис. 5.4. Диаграмма классов модели банкомата после выбора стереотипа для класса Транзакция Банкомата

чения стереотипа классу Транзакция банкомата текст данного стереотипа в угловых скобках появится выше имени данного класса (рис. 5.4).

Для отдельного класса можно уточнить также и другие его свойства, доступные для редактирования на вкладках Attributes, Operations, Templates, Associations окна спецификации свойств этого класса. Например, на вкладке Attributes можно создать новый атрибут (поле) для рассматриваемого класса, который будет иметь тип (bool, char, double, float, int, long, short, string, unsigned int, unsigned long и unsigned short), имя, начальное значение, имя стереотипа, возможность указания спецификатора static. По умолчанию видимость нового атрибута (поля) класса будет иметь спецификатор доступа «private» (рис. 5.5).

Также для класса можно создать новую функцию – метод класса, используя вкладку Operations диалогового окна свойств класса. При этом указываются имя, имя стереотипа, тип возвращаемого значения и список параметров (рис. 5.6). Можно указать спецификаторы const, static и принадлежность операции к абстрактным (виртуальным). По умолчанию видимость нового метода класса будет иметь спецификатор доступа «public».

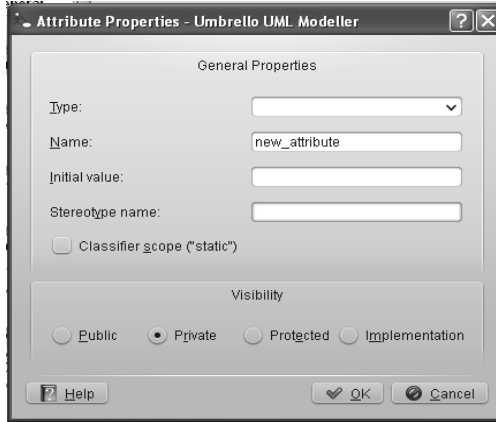


Рис. 5.5. Добавление нового атрибута класса

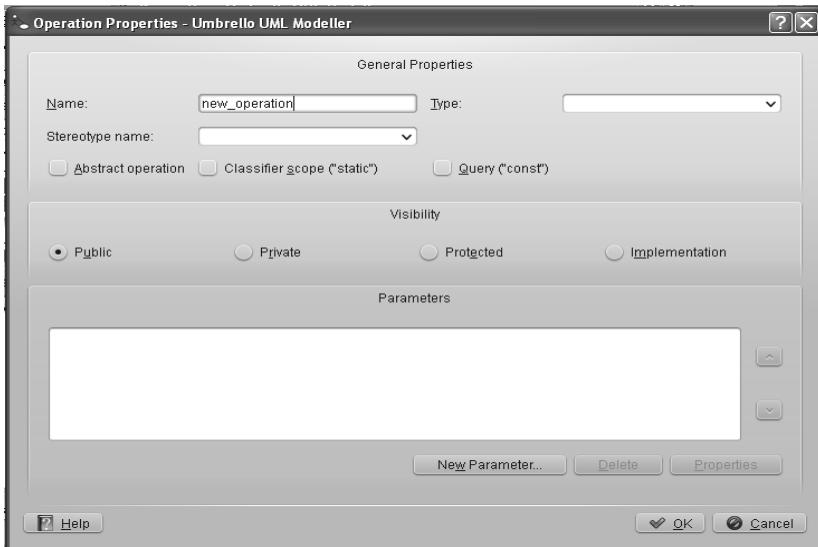


Рис. 5.6. Добавление нового метода класса

На вкладке Templates диалогового окна свойств класса можно указать параметры добавленного в описание класса шаблона (рис. 5.7). На вкладке Associations задаются ассоциации данного класса с другими элементами диаграммы. Вкладки Display, Color и Font позволяют управлять отображением класса, например на вкладке Display можно

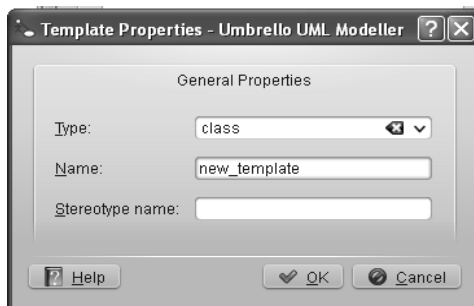


Рис. 5.7. Добавление шаблона

указать, будут ли определённые элементы класса (поля, методы, шаблоны, стереотипы и другое) отображаться на поле диаграммы классов.

Для того, чтобы специфицировать класс как абстрактный, т.е. не имеющий экземпляров, следует на этой же вкладке выставить отметку в свойстве **Abstract** (Абстрактный). Применительно к рассматриваемой модели для класса «Транзакция банкомата» следует отметку для свойства **Abstract** оставить пустой. Следует заметить, что для предотвращения потери информации о разрабатываемой модели и результатов редактирования свойств её графических элементов необходимо периодически сохранять модель во внешнем файле. Для этого следует выполнить операцию главного меню: **File->Save** (Файл ->Сохранить) или нажать комбинацию клавиш: **Ctrl+S**. Для этой же цели служит соответствующая кнопка на стандартной панели инструментов.

Разработка диаграммы классов модели банкомата.

На разрабатываемой диаграмме классов выбран текстовый способ изображения стереотипов классов, при котором стереотип записывается в угловых кавычках выше имени соответствующего класса.

Выделить все графические элементы на диаграмме классов, так же как и на любой другой диаграмме модели, можно с помощью выполнения операции главного меню: **Edit->Select All** (Редактирование->Выделить все) или с помощью комбинации клавиш **Ctrl+A**. Продолжая разработку модели банкомата, добавим на диаграмму второй класс с именем «Контроллер Банкомата», для которого в окне спецификации свойств выберем стереотип **control** (управляющий класс), а в качестве документации введём текст: «Реализует логику функционирования банкомата». При этом атрибуты и операции у данного класса будут отсутствовать. Соответствующий фрагмент диаграммы классов после добавления управляющего класса Контроллер Банкомата будет иметь следующий вид (рис. 5.8).

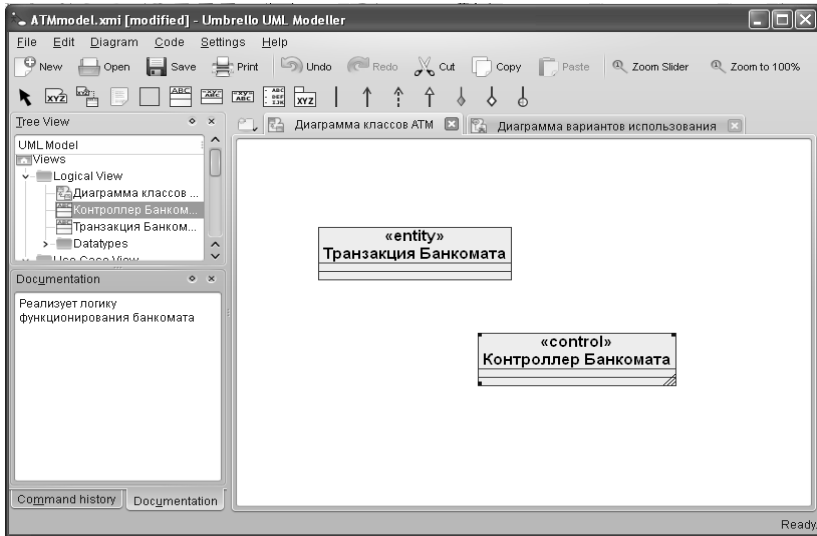


Рис. 5.8. Диаграмма классов после добавления класса «Контроллер Банкомата»

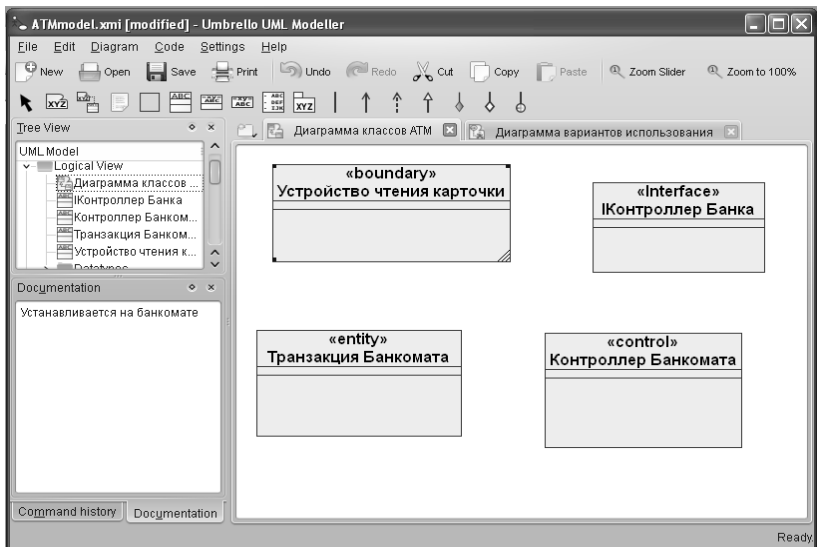


Рис. 5.9. Классы модели банкомата

Продолжая разработку модели банкомата, добавим на диаграмму третий класс с именем «Устройство чтения карточки», для которого в окне спецификации свойств выберем стереотип «boundary» (граничный класс). Применение этого стереотипа означает, что данный класс находится на границе моделируемой системы, в качестве которой рассматривается модель банкомата. После этого в секцию документации данного класса можно ввести поясняющий текст: «Устанавливается на банкомате». Далее следует добавить интерфейс с именем «Контроллер Банка», для него стереотип «Interface» (Интерфейс) будет установлен по умолчанию, означающий, что банкомат пользуется услугами Банка при обработке своих транзакций. Заметим, что первой буквой в имени этого класса является английское «I», которое служит в языке UML для указания интерфейса. Соответствующий фрагмент диаграммы классов после добавления на неё классов «Устройство чтения карточки» и «Контроллер Банка» будет иметь вид, представленный на рис. 5.9.

5.2. СРЕДСТВА ПРОЦЕССНО-ОРИЕНТИРОВАННОГО (ФУНКЦИОНАЛЬНОГО) МОДЕЛИРОВАНИЯ

Для выполнения лабораторных работ по функциональному моделированию предлагается использовать редактор диаграмм Dia. Он был выбран, так как удовлетворяет два ключевых требования: бесплатность и кроссплатформенность.

Dia – кроссплатформенный свободный редактор диаграмм, часть GNOME Office, но может быть установлен независимо. Он может быть использован для создания различных видов диаграмм: блок-схем алгоритмов программ, древовидных схем, статических структур UML, баз данных, диаграмм сущность-связь, радиоэлектронных элементов, потоковых диаграмм, сетевых диаграмм и других.

Dia расширяем новыми наборами объектов, которые описываются с помощью файлов в формате, основанном на XML.

Он обладает такими возможностями, как:

- поддержка диаграмм потоков, структурных диаграмм и т.д.;
- экспорт в Postscript;
- загрузка и сохранение в формате XML;
- возможность описания новых объектов;
- установка свойств по умолчанию для добавляемых объектов;
- изменение цвета шрифта и заливки блоков.

К нему доступны следующие надстройки:

– AutoDia – автоматическое создание UML-схем из программно-го кода;

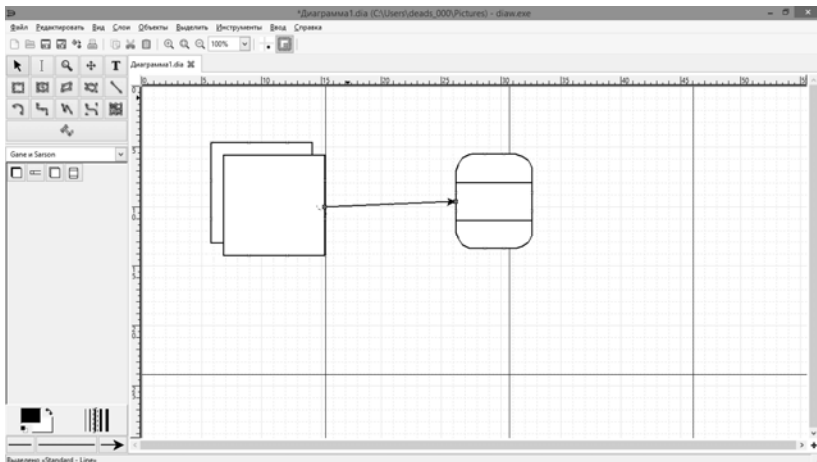


Рис. 5.10. Пример интерфейса Dia

– Dia2Code – автоматическое преобразование UML-схем в программный код.

Интерфейс Dia (рис. 5.10) разбит на две половины: одна содержит поле для диаграммы, другая – набор инструментов.

Базовые элементы диаграмм, такие как стрелки и текст, доступны всегда, элементы же специфичные для определённой диаграммы выбираются из соответствующего набора.

Для построения DFD-диаграмм используется стандартный набор под названием «Gane и Sarson», представленный на рис. 5.11.

Данный набор состоит из следующих элементов: «другая внешняя сущность», «хранилище данных», «внешняя сущность», «обработчик».

Для построения IDEF0-диаграмм используется набор «SADT/IDEF0» (рис. 5.12).

В наборе «SADT/IDEF0» есть следующие элементы: «Направление потока активности/данных», «блок активность/данные», «плавающая метка».



Рис. 5.11. Набор «Gane и Sarson»



Рис. 5.12. Набор «SADT/IDEF0»

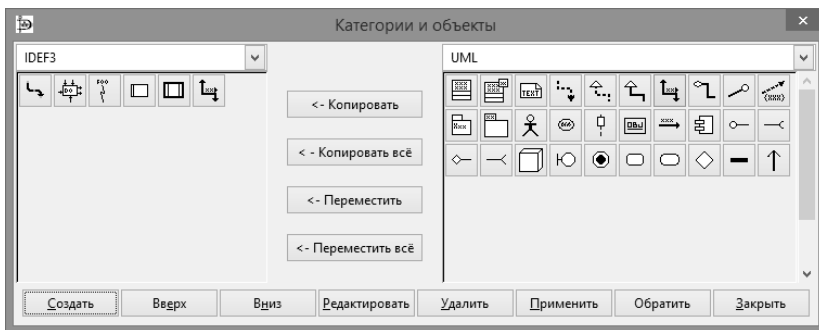


Рис. 5.13. Создание набора для IDEF3

В стандартные наборы Dia не входит набор, позволяющий строить IDEF3-диаграммы, но его можно создать самим, для этого нужно сделать следующее:

- нажимаем в меню: «Файл -> категории и объекты»;
- в открывшемся окне нажимаем кнопку «Создать»;
- откроется ещё одно окошко, в котором выбираем пункт «Название категории» и вписываем туда «IDEF3», нажимаем «OK».

Получится пустой набор, далее нужно его заполнить, для этого делаем следующее:

- для начала справа выбираем категорию SADT/IDEF0, там будет 3 элемента. Нажимаем кнопку «копировать все» (смотрите не нажмите случайно «Переместить всё»);
- теперь справа выбираем категорию «RE-Джексона» и копируем фигуру «прямоугольник» с одинарной линией слева;
- теперь выбираем категорию «Блок-схема» и оттуда добавляем прямоугольник с линиями по бокам
- и, наконец, выбираем категорию UML и добавляем оттуда чёрную изогнутую линию.

В результате получится набор, указанный на рис. 5.13.

После этого полученный набор можно использовать для построения диаграмм IDEF3.

Задания для самопроверки

1. Постройте диаграмму прецедентов для терминала выдачи билетов электронной очереди.
2. Постройте IDEF0-диаграмму для терминала выдачи билетов электронной очереди.

ЗАКЛЮЧЕНИЕ

Очевидно, что невозможно собрать в рамках одного издания все известные модели и методы системного анализа, количественные и качественные методы моделирования и проектирования, позволяющие в абсолютной полноте представить теорию информационных процессов и систем. Поэтому авторами данного учебника была сделана попытка выделить наиболее принципиальные теоретические аспекты анализа и синтеза информационных систем и процессов в них.

Рассмотренные основные понятия по своей сути отражают перенос понятийного аппарата общей теории систем на область информационных процессов, учитывая, что информационные процессы и системы – ещё очень «молодая», только начинающая развиваться предметная область, которая в то же время охватывает практически все сферы деятельности современной экономики, техники и технологий.

Математические методы моделирования систем позволяют понять общие принципы и подходы к формализации процессов в информационных системах, освоив которые специалист должен научиться выявлять физические особенности и условия протекания информационных процессов с целью выбора математического аппарата для их формального отображения и последующего исследования.

Качественные и количественные методы представления процессов и систем на сегодняшний день известны в таком многообразии, что стремиться освоить всё, значит стремиться объять необъятное. В учебнике рассмотрены и предлагаются обучаемым наиболее доступные, широко известные и часто применяемые на практике методы описания информационных систем и процессов.

Методы анализа информационных систем и процессов рассмотрены в зависимости от условий, в которых проводится оценка, причём с учётом различных факторов, влияющих на эти условия.

Примеры инструментальных средств моделирования и анализа информационных процессов и систем дают возможность разобраться в особенностях технологических подходов к системному анализу в этой области.

СПИСОК ЛИТЕРАТУРЫ

1. **Ананьин, В.** Интранет как инструмент корпоративного управления / В. Ананьин // Системы управления базами данных. – 1997. – № 4. – С. 80 – 87.
2. **Анфилатов, В. С.** Системный анализ в управлении : учебное пособие / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин. – Москва : Финансы и статистика, 2002. – 368 с.
3. **Базы данных** / А. Д. Хомоненко [и др.]. – СПб. : Корона принт, 2000. – 416 с.
4. **Волкова, В. Н.** Теория систем и системный анализ в управлении организации [Электронный ресурс]. – М. : Финансы и статистика, 2009. – 848 с. – Режим доступа : <http://www.elanbook.ru>
5. **Волкова, В. Н.** Теория систем : учеб. пособие / В. Н. Волкова, А. А. Денисов. – Москва : Высш. шк., 2006. – 511 с.
6. **Гаврилова, Т. А.** Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – Санкт-Петербург : Питер, 2000. – 384 с.
7. **Гради, Б.** Объектно-ориентированный анализ и проектирование с примерами на C ++ / Б. Гради. – 2-е изд. – Москва : БИНОМ ; Санкт-Петербург : Нев. диалект, 1998. – 560 с.
8. **Грей, Д.** Управление данными: Прошлое, Настоящее и Будущее / Д. Грей // Системы управления базами данных. – 1998. – № 3. – С. 71 – 80.
9. **Системный анализ в информационных технологиях** [Электронный ресурс] : учебное пособие / Ю. Ю. Громов, О. Г. Иванова, М. А. Ивановский, С. В. Данилкин, Д. П. Швец. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 176 с.
10. **Гусева, А. И.** Технология межсетевых взаимодействий. NttWag- Unix-Windows-Internet / А. И. Гусева. – Москва : Диалог-МИФИ, 1997. – 272 с.
11. **Дейт, К. Д.** Введение в системы баз данных / К. Д. Дейт. – Москва : Диалектика, 2002. – 1071 с.
12. **Джексон, П.** Введение в экспертные системы / П. Джексон. – Москва : Вильямс, 2001.
13. **Елисеев, В.** Введение в Интранет / В. Елисеев, Г. Ладыженский // Системы управления базами данных. – 1996. – № 5. – С. 19 – 43.
14. **Игумнов, Е.** Методология построения корпоративных информационных систем на основе технологии EJB [Электронный ресурс]. – Режим доступа : [http // www.ejbcorba. euro. ru.](http://www.ejbcorba.euro.ru)

15. **Йордон, Э.** Структурные модели в объектно-ориентированном анализе и проектировании / Э. Йордон, К. Аргила. – Москва : Лори, 1999. – 264 с.

16. **Каляное, Г. Н.** CASE-технологии: Консалтинг в автоматизации бизнес-процессов / Г. Н. Калянов. – Москва : Горячая линия ТЕЛЕКОМ, 2002. – 320 с.

17. **Козлов, В. А.** Открытые информационные системы / В. А. Козлов. – Москва : Финансы и статистика, 1999. – 223 с.

18. **Львов, В.** Создание систем поддержки принятия решений на основе хранилищ данных / В. Львов // Системы управления базами данных. – 1997. – № 3. – С. 30 – 40.

19. **Назаров, С. В.** Компьютерные технологии обработки информации. – Москва : Финансы и статистика, 1999. – 249 с.

20. **О'Коннор, Дж.** Искусство системного мышления: необходимые знания о системах и творческом подходе к решению проблем / Дж. О'Коннор, И. Макдермот. – Москва : Альпина Бизнес Букс, 2006. – 256 с.

21. **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – СПб. : Питер, 1999. – 668 с.

22. **От классической логики к логическому программированию /** Тейз [и др.] ; пер. с фр. – Москва : Мир, 1990. – 432 с.

23. **Практикум по пакетам прикладных программ /** С. В. Назаров [и др.]. – Москва : Финансы и статистика, 1999. – 320 с.

24. **Поспелов, Г. С.** Искусственный интеллект – основа новой информационной технологии / Г. С. Поспелов. – Москва : Наука, 1988. – 280 с.

25. **Робсон, М.** Практическое руководство по реинжинирингу бизнес-процессов / М. Робсон, Ф. Уллах. – Москва : Аудит Юнити, 1997. – 224 с.

26. **Саймон, А. Р.** Стратегические технологии баз данных: менеджмент на 2000 год / под ред. М. Р. Когаловского. – Москва : Финансы и статистика, 1999. – 479 с.

27. **Советов, Б. Я.** Информационная технология. – Москва : Высшая школа, 1994. – 368 с.

28. **Советов, Б. Я.** Информационные технологии / Б. Я. Советов, В. Цехановский. – Москва : Высшая школа, 2004. – 263 с.

29. **Советов, Б. Я.** Базы данных: теория и практика / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – Москва : Высшая школа, 2004. – 463 с.

30. **Советов, Б. Я.** Теоретические основы автоматизированного управления / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – Москва : Высшая школа, 2005. – 463 с.

31. **Суздалов, Е. Г.** Конспект лекций по дисциплине: ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ [Электронный ресурс]. – СПб., 2010. – Режим доступа : http://window.edu.ru/window_catalog/files/r67923/Suzdalov_systems_analysis.pdf

32. **Статические** и динамические экспертные системы / Э. В. Попов [и др.]. – Москва : Финансы и связь, 1996. – 352 с.

33. **Теллин, С.** Интранет и Адаптивные Инновации: переход от управления к координации в современных организациях / С. Теллин // Системы управления базами данных. – 2003. – № 5–6. – С. 43 – 47.

34. **Технологии** искусственного интеллекта : в 2 ч. / Б. Я. Советов [и др.]. – СПб. : Техномедиа : Элмор, 2007. – Ч. 1. – 262 с.

35. **Ульман, Д. Д.** Введение в системы баз данных / Д. Д. Ульман, Д. Уидром. – Москва : Лори, 2000. – 374 с.

36. **Хейг, Крикорян.** Введение в объектно-ориентированную системную инженерию / Крикорян Хейг // Открытые системы. – 2003. – № 11. – С. 35 – 41.

37. **Шапот, М.** Интеллектуальный анализ данных в системах поддержки принятия решений / М. Шапот // Системы управления базами данных. – 1998. – № 1. – С. 41 – 49.

38. **Швецов, А. Н.** Распределенные интеллектуальные информационные системы / А. Н. Швецов, С. А. Яковлев. – СПб. : СПбГЭТУ «ЛЭТИ», 2003. – 318 с.

39. **Шеховцов, О. И.** Интеллектуальные средства поддержки принятия управленческих решений / О. И. Шеховцов, В. Д. Чертовской, Б. М. Шифрин. – СПб. : СПбГЭТУ «ЛЭТИ», 2000. – 59 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ	7
1.1. Историческая справка	7
1.2. Терминология теории систем.....	10
1.3. Подходы к классификации систем	18
1.4. Свойства (закономерности) систем	22
1.5. Особенности системного подхода и системного анализа	28
1.6. Системные понятия информационного процесса, информационной технологии, информационной системы	37
2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ	41
2.1. Общие подходы к математическому моделированию систем	41
2.2. Каноническое представление информационной системы	45
2.2.1. Критерии качества информационной системы	47
2.2.2. Критерии эффективности функционирования информационной системы	48
2.3. Теоретико-множественные модели информационных систем	51
2.3.1. Система как отношение на абстрактных множествах	52
2.3.2. Временные, алгебраические и функциональные системы	54
2.4. Моделирование систем сетями Петри	59
2.5. Моделирование аппаратного обеспечения вычислительных систем	68
2.6. Принятие решений как задача системного анализа	70
3. МЕТОДЫ И МОДЕЛИ ОПИСАНИЯ (ПРЕДСТАВЛЕНИЯ) СИСТЕМ	75
3.1. Качественные методы описания систем	75
3.2. Модели процессов и систем на основе декомпозиции и агрегирования	81
3.3. Объектно-ориентированное моделирование информационных процессов и систем диаграммами UML	85
3.4. Процессно-ориентированное моделирование информационных процессов и систем	100
3.5. Количественное описание информационных процессов и систем	116

4. МЕТОДЫ АНАЛИЗА (ОЦЕНКИ) ИНФОРМАЦИОННЫХ СИСТЕМ	125
4.1. Многокритериальная оценка систем в условиях определённости	125
4.2. Оценка сложных систем на основе теории полезности	132
4.2.1. Оценка сложных систем в условиях риска на основе теории полезности	135
4.2.2. Оценка сложных систем в условиях неопределённости	136
4.3. Анализ информационных систем на основе комплексного применения качественных и количественных системных методов	140
4.4. Методы статистической оценки информационных систем	144
5. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СИСТЕМНОГО АНАЛИЗА И МОДЕЛИРОВАНИЯ	154
5.1. Среда разработки объектно-ориентированных моделей Umbrello UML Modeller	154
5.2. Средства процессно-ориентированного (функционального) моделирования	163
ЗАКЛЮЧЕНИЕ	166
СПИСОК ЛИТЕРАТУРЫ	167

Учебное издание

ГРОМОВ Юрий Юрьевич,
ДИДРИХ Валерий Евгеньевич,
ИВАНОВА Ольга Геннадьевна,
ОДНОЛЬКО Валерий Григорьевич

ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Учебник

Редактор Т. М. Глинкина
Инженер по компьютерному макетированию И. В. Евсева

ISBN 978-5-8265-1352-1



Подписано в печать 10.11.2014
Формат 60 × 84 / 16. 10,0 усл. печ. л.
Тираж 100 экз. Заказ № 524

Издательско-полиграфический центр
ФГБОУ ВПО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Тел. 8(4752) 63-81-08
E-mail: izdatelstvo@admin.tstu.ru