

1. Построение макромоделей АСОИУ на предпроектной стадии ее разработки.

Одной из важнейших целей предпроектного анализа создаваемой АСОИУ является построение ее макромоделей. Такая макромодель состоит из 4-х матриц следующего вида:

- а) цели системы управления – (матрица Ф0),
- б) цели системы управления – функции (матрица Ф1),
- в) функции системы управления – задачи (матрица Ф2),
- г) задачи – информационные потребности (матрица Ф3).

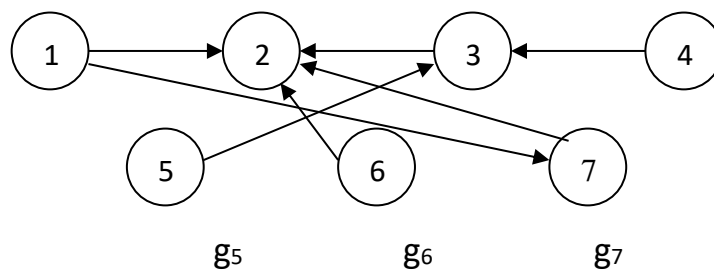
Для построения этих матриц может применяться **экспертный метод**.

Начинается построение макромоделей с построения матрицы «цели – цели» (Ф0).

Пусть путем экспертного опроса выявлены цели проектируемой АСОИУ

Номер цели	Формулировка цели	Обеспечивает цели
1	Увеличить выпуск _____продукции до 10 тыс. шт.	2 и 7
2	Повысить рентабельность производства	–
3	Обеспечить ритмичность работы производства	2
4	Повысить обоснованность планов	3
5	Обеспечить выполнение заказов в срок	3
6	Упорядочить потребление материалов	2
7	Повысить качество продукции	2

Полученные данные можно представить в виде графа целей G1 и матрицы целей Ф0.



Матрица целей Ф0

цель \ цель	1	2	3	4	5	6	7
1		1					1
2							
3		1					
4			1				
5			1				
6		1					
7		1					

Исходный граф целей можно упорядочить, расположив вершины по уровням иерархии, имея в виду, что на первом уровне находятся цели, для достижения которых может быть осуществлено непосредственно, на втором уровне цели – которые могут быть достигнуты после достижения целей на первом уровне, и т.д. Такой граф имеет для нашего примера следующий вид (см. рис. 4.6.2):

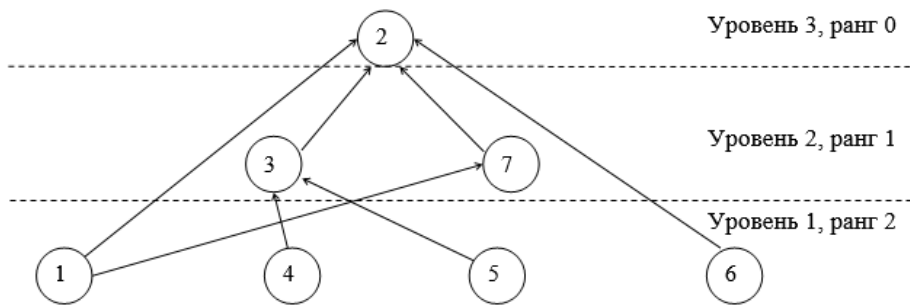


Рис. 4.6.2. Граф взаимосвязи целей

Граф целей позволяет дать предварительную оценку их значимости. Значимость цели определяется ее уровнем в иерархии целей, а также количеством и уровнем целей, которые обеспечиваются достижением оцениваемой цели. На основе этих положений предварительная оценка значимости целей может быть представлена следующим простым соотношением

$$\alpha_j = \frac{P_j(\sum_{s=1}^n \delta_{js} P_s + P_j)}{\sum_i P_i(\sum_{s=1}^n \delta_{is} P_s + P_i)},$$

где P_j – уровень j -й цели,

$$\delta_{js} = \begin{cases} 1, & \text{если } j\text{-я цель обеспечивает } s\text{-ю цель} \\ 0, & \text{если не обеспечивает.} \end{cases}$$

Для нашего примера имеем:

$g_1 : P_1(P_2+P_7+P_1)=1(3+2+1)=6$	$\alpha_1=0,13$
$g_2 : P_2P_2=3 \cdot 3=9$	$\alpha_2=0,20$
$g_3 : P_3(P_2+P_3)=2(3+2)=10$	$\alpha_3=0,22$
$g_4 : P_4(P_3+P_4)=1(2+1)=3$	$\alpha_4=0,07$
$g_5 : P_5(P_3+P_5)=1(2+1)=3$	$\alpha_5=0,07$
$g_6 : P_6(P_2+P_6)=1(3+1)=4$	$\alpha_6=0,09$
$g_7 : P_7(P_2+P_7)=2(3+2)=10$	$\alpha_7=0,22$
$\sum_{j=1}^7 P_j(\sum_{s=1}^7 \delta_{js} P_j + P_j) = 45$	$\sum_{j=1}^7 \alpha_j = 1$

Для обеспечения выявленных целей АС должна реализовывать определенные функции управления. Пусть в результате опроса специалистов получена таблица (Ф1) вида «цели – функции» Ф1. Она показывает,

какие функции обеспечивают достижение каких целей. Для рассматриваемого примера эта матрица имеет следующий вид:

Матрица «цели – функции» Ф1

Функции (i)	Цели (j)							
	Уровень 3	Уровень 2		Уровень 1				γ _i
	2	3	7	1	4	5	6	
1. Перспективное планирование				1	1	1		0,06
2. Текущее планирование		1			1	1		0,08
3. Подготовка производства		1	1	1				0,13
4. Диспетчеризация		1			1	1	1	0,10
5. Материально-техническое обеспечение		1	1	1	1	1	1	0,18
6. Планирование загрузки производственных мощностей	1	1	1	1	1	1	1	0,17
7. Планирование и учет труда и заработной платы	1	1		1	1	1		0,2
8. Бухучет и финансы	1							0,04
9. Статистический учет	1							0,04
Σ	4	6	3	5	6	6	3	1,0

Анализ этой матрицы позволяет, прежде всего, ранжировать функции по их значимости, учитывая значимость и количество обеспечиваемых целей. Оценку значимости функций γ_i можно представить в виде:

$$\gamma_i = \frac{\sum_j \delta_{ij} \alpha_j}{\sum_i \sum_j \delta_{ij} \alpha_j}$$

где α_j – оценка значимости целей,

$$\delta_{ij} = \begin{cases} 1, & \text{если функция с номером } i \text{ обеспечивает цель с номером } j; \\ 0, & \text{если функция с номером } i \text{ не обеспечивает цель с номером } j. \end{cases}$$

Из матрицы Ф1 по коэффициенту значимости функций γ_i следует, какие функции надо автоматизировать в первую очередь, вторую и т.д.

Матрица Ф1 позволяет также отметить такую особенность организации, как высокая или низкая концентрация функций по отношению к поставленным целям. Так, в рассмотренном примере достижение цели g_3 связано с эффективным выполнением шести функций, что требует определенных усилий по их координации.

Для обеспечения указанных функций в системе управления должны решаться определенные задачи управления. Для этого необходимо построить матрицу «функции – задачи» Ф2. Она показывает, реализация какого набора задач обеспечивает выполнение выбранных функций.

Задачи (k)	Функции (i)									
	1	2	3	4	5	6	7	8	9	μ_k
1. Повышение уровня расчетной обоснованности плановых показателей		1				1			1	0,14
2. Упорядочение нормативов			1		1				1	0,18
3. Анализ ритмичности производства		1		1						0,14
4. Расчет динамики заработной платы							1	1		0,05

5. Анализ себестоимости					1		1	1		0,12
6. Экономическое обоснование конструкции изделия					1					0,06
7. Лимитирование оппуска материалов					1					0,07
8. Составление материальных балансов					1			1		0,09
9. Расчет рентабельности		1						1		0,08
10. Учет затрат на капитростроительство	1							1		0,07
Σ	1	3	1	1	5	1	2	5	2	1,0

Анализ этой матрицы может позволить определить важность задач. Для этого введем коэффициент важности задачи μ_k .

Коэффициент важности задач μ_k определяют по формуле

$$\mu_k = \frac{\sum_i \delta_{ki} \gamma_i}{\sum_i \sum_k \delta_{ki} \gamma_i},$$

где γ_i – значимость i -й функции

$$\delta_{ki} = \begin{cases} 1, & \text{если } k\text{-я задача обеспечивает выполнение } i\text{-й функции,} \\ 0, & \text{в противном случае} \end{cases}$$

Теперь представляется возможным построить полную матрицу смежности целей и задач и уточнить оценки их значимости в организации и управлении.

Итак, преобразования матрицы приводят к распределению задач по уровням p^* , приведенным в верхней строке табл. \tilde{M} . При этом часть задач выдвигается на достаточно высокий уровень и становится равнозначной поставленным целям.

Уточненная оценка значений целей и задач α^* основывается на принципах, положенных в основу определения α .

Для упрощения последующих записей пронумеруем столбцы и строки матрицы \tilde{M} , положив номера строк $i=j$ для g и $i=k+7$ для a_k и аналогично – номера столбцов $s=j$ для g и $s=k+7$ для a_k . Тогда в этих новых обозначениях формула для вычисления α_i^* имеет вид:

$$\alpha_i^* = \frac{P_i^* (\sum_s \delta_{is} v_s P_s^* + v_i P_i^*)}{\sum_i P_i^* (\sum_s \delta_{is} v_s P_s^* + v_i P_i^*)},$$

где $v_s = \{\alpha_i^*, \mu_k\}$, δ_k – элементы матрицы \tilde{M} . Значения α_i^* приведены в таблице.

Если какая-то строка содержит лишь 0, это означает, что данная цель (задача) является конечной (результатирующей). Если же 0 содержит весь столбец, то соответствующую задачу следует считать исходной. Нулевые строка и столбец одного и того же номера означают, что данная задача одновременно и исходная, и результирующая, т.е. автономная. В этом случае целесообразно более детально проанализировать ее содержание. Часто есть основание полагать, что задача надуманная и никому не нужна, редко – это некая особая цепь.

P_S^*	4	7	6	5	5	4	5	2	1	3	2	3	2	2	3	4	3		
$\alpha_j; \mu_k$	0,13	0,2	0,22	0,07	0,07	0,09	0,22	0,14	0,08	0,14	0,05	0,12	0,06	0,07	0,09	0,08	0,07		
$i \backslash s$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}		α_i
g_1		1					1											2	0,084
g_2																		0	0,042
g_3		1																1	0,106
g_4			1															1	0,054
g_5			1															1	0,054
g_6		1																1	0,046
g_7		1																1	0,060
a_1		1	1	1	1							1						5	0,053
a_2	1	1	1		1	1	1	1			1		1	1				10	0,041
a_3			1	1	1		1									1		5	0,08
a_4	1	1		1						1		1				1		6	0,031
a_5	1	1	1	1	1	1										1		7	0,056
a_6			1		1	1				1					1			5	0,037
a_7			1		1	1						1					1	5	0,036
a_8		1	1		1	1										1		5	0,079
a_9		1	1	1	1													4	0,097
a_{10}		1		1												1		3	0,044
	3	11	10	6	8	7	3	1	0	2	1	3	1	1	1	5	1		1.0

Рис. 4.6.2. Матрица смежности «Цели – задачи» \tilde{M}

Сравнение состава задач с их распределением по подразделениям оргсистемы позволяет выявить не решаемые задачи и установить, в какой мере квалификация и оплата исполнителей совпадают с важностью решаемых ими задач. Показатель α^*_i дает возможность выявить узловые задачи, решение которых имеет наибольшее значение для успешного функционирования системы. В данном примере это задачи a_5 и a_6 .

После анализа матрицы цели – задачи можно переходить к построению матрицы ФЗ «задачи – информация». Ее анализ позволяет установить степень загруженности подразделений, высказать рекомендации по их штатной численности, предпочтительной схеме распределения решаемых задач и целесообразном уровне автоматизации. Эта матрица формируется следующим образом:

- 1) устанавливается информация, получаемая тем или иным функциональным или линейным подразделением;
- 2) устанавливается информация, формируемая в функциональных или линейных подразделениях;
- 3) определяются виды информации, используемой в системе управления;
- 4) определяются необходимые для обеспечения данных задач виды информации;
- 5) определяется, какое количество каждого вида информации в принятый за расчетный период необходимо для обеспечения решения рассматриваемых задач.

Анализ полученной таблицы позволяет установить:

- 1) максимальное количество информации определенного вида в течение установленного периода $F_{\text{кпик}}$;
- 2) суммарное количество информации, необходимое для решения каждой задачи F_i ;
- 3) вид информации, требуемой для решения соответствующей i -й задачи;
- 4) виды информации, которые не используются непосредственно в решении задач;
- 5) задачи, не обеспеченные информацией.

Полученные данные можно представить в виде таблицы ФЗ «задачи – информация»

Вид информации	Цели и задачи (i)					$F_{\text{кпик}}$
	1	2	...	16	17	
1. цена готовой продукции	2	1		1		
2. расчет трудоемкости	3	2		4		
3. справки	5			6		
4. спецификация	10			8		
⋮						
Общее количество информации, обеспечивающее решение i -й задачи F_i [Т бит/сутки]						

Если φ – количество информации, которую может обработать один человек в заданный период, то $\varphi = 10^5 \div 10^6$ (бит/сутки)

Пусть F_0 – количество информации, циркулирующей в самой организационной системе, а $F_{\text{вн}}$ – внешняя информация, которая вносится в процессе управления извне. Тогда можно определить максимальную численность людей в организации

$$N_{\text{max}} = \frac{b \sum_i F_i}{\varphi}, \text{ где } b = \frac{2F_0 + F_{\text{вн}}}{F_0 + F_{\text{вн}}}.$$

Этот коэффициент b учитывает тот факт, что любая производственная информация должна быть кем-то в организации использована, а используемая – кем-то произведена.

На основании матриц «цели – цели», «цели – функции», «функции – задачи», «задачи – информация» можно построить обобщенную логико-информационную модель проектируемой АСОИУ.

Эту модель можно представить в виде трехдольного ориентированного графа $G_0=(Z, W)$, где Z – множество вершин, состоящее из четырех подмножеств: $Z=Z_1 \cup Z_2 \cup Z_3 \cup Z_4$, $Z_1 \cap Z_2 \cap Z_3 \cap Z_4 = \emptyset$, где Z_1 – подмножество целей, Z_2 – подмножество функций, Z_3 – подмножество задач, Z_4 – подмножество информационных элементов; а W – множество дуг, показывающих информационную взаимосвязь между элементами множества Z . Если дуги множества W отражают структуру информации, то граф G_0 превращается в мультиграф.

Для упрощения этого графа иногда выделяют из него подграф, относящийся к целям.

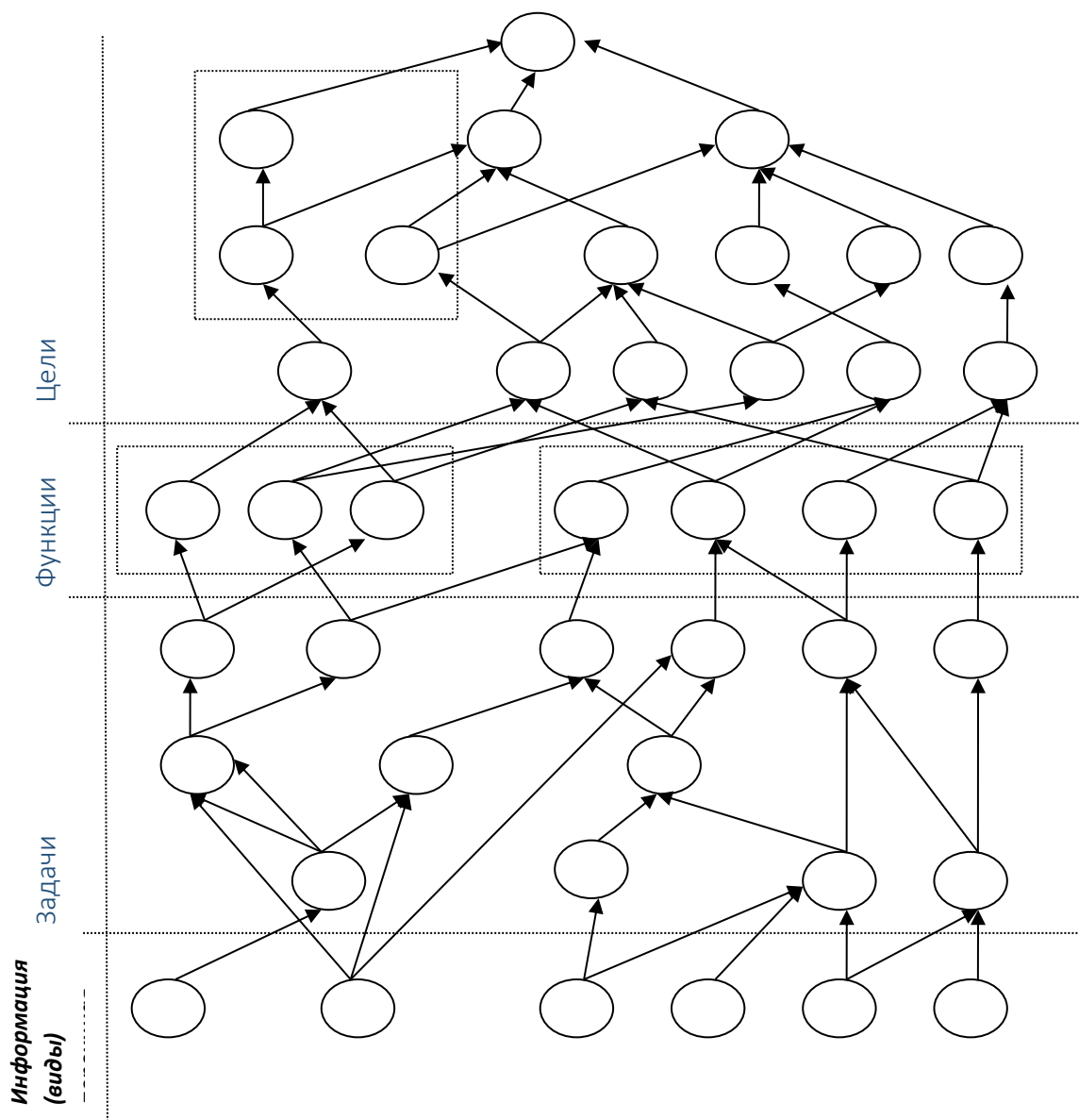
В результате получают логико-информационный граф, показывающий связь информационных элементов, задач и функций управления. При этом функции управления отождествляют с функциональными подсистемами проектируемой АСОИУ. Пример такого графа приведен

Построение схемы информационно-логической взаимосвязи задач позволяет:

- 1) определить последовательность решения задач;
- 2) определить группы задач, решение которых может производиться параллельно;
- 3) уточнить объем нормативной информации;
- 4) определить необходимые сроки хранения исходных, промежуточных и результирующих данных.

Построение информационной схемы задач производится на основе разбиения их на классы.

Обобщенная логико-информационная модель проектируемой АСУ



Определение 1. Информация, поступающая для решения задач извне, называется первичной.

Определение 2. Множество задач $(\ell+1)$ -го класса $L^{\ell+1}$ определяется как множество, для решения которого необходимы и достаточны первичная информация и результаты решения задач $L^1, L^2, \dots, L^{\ell}$. Среди задач класса $(\ell+1)$ нет ни одной задачи, результаты решения которой используются при решении других задач этого же класса.

Определение 3. Для каждой задачи из множества $L^{\ell+1}$ существует хотя бы одна задача, во множестве L^{ℓ} , результаты решения которой использовались бы для решения $L^{\ell+1}$.

Из предложенных определений следует, что к первому классу L^1 относятся все те задачи, для решения которых достаточна первичная информация. Если результаты решения i -й задачи используются при решении какой-либо задачи ℓ -го класса, а сама i -я задача решается только на основе первичной информации, а также результатов решения задач первого класса, то она относится непосредственно к $(\ell-1)$ -му классу.

Итак, пусть заданы неупорядоченное множество задач $L=\{L_i\}$ и информационные связи между ними $U=\{U_k\}$, образующие исходный граф $G=\{L, U\}$. Ставится задача: упорядочить множество L задач Z на подмножество классов, в которых задачи не связаны между собой смежной информацией.

Для решения поставленной задачи зададим граф $G=\{L, U\}$ в виде матрицы смежности $H=(L, U)$ размерности $n \times m$, у которой номера строк соответствуют номерам задач (вершинам), а номера столбцов – номерам дуг, показывающих информационную связь между задачами. Значения элементов S_{ij} матрицы H следующие:

$$S_{ij} = \begin{cases} 1, & \text{если задача (вершина) } L_i \text{ есть начало дуги } U_j \\ 2, & \text{если задача (вершина) } L_i \text{ есть конец дуги } U_j \\ 0, & \text{если задача } L_i \text{ не принадлежит дуге } U_j \end{cases}$$

Анализируя эту матрицу, можно осуществить

формирование классов задач. Алгоритм такого анализа состоит в последовательном просмотре строк матрицы и выявлении таких строк, которые содержат только 0 и 1. На очередном k -ом шаге все выявленные строки отвечают задачам k -го класса ($k=1, 2, \dots$). После поиска таких строк и их исключения из матрицы вместе с соответствующими столбцами, где стояли единицы, переходят к следующему $(k+1)$ -му шагу и процесс повторяется до получения нулевой матрицы.

Модульный принцип. (Общая инфа для след билетов)

Проектирование АСОИУ с использованием модульного принципа связано с созданием программного и информационного обеспечения АСОИУ из некоторого множества относительно независимых частей или модулей обработки данных, которые имеют информационные взаимосвязи, определенные таким образом, что каждый модуль не получает информации о внутреннем содержании других модулей, кроме той, которая содержится в спецификациях интерфейса.

Применение модульного принципа проектирования АСОИУ позволяет свести эти проектирование к синтезу функционально независимых отдельных частей (модулей), совместно выполняющих заданные функции системы с требуемой эффективностью. При этом внутреннее содержание модулей может быть не известно.

Модульное проектирование АСОИУ обладает рядом преимуществ:

- упрощается процесс разработки и отладки программного и информационного обеспечения АСОИУ;
- упрощается последующая модификация системы, так как модульные программы могут быть улучшены путем простой замены отдельных модулей, которые функционально эквивалентны, но имеют лучшие системные характеристики;
- улучшается организация управляющих программ;
- появляются возможности внедрения передовых методов разработки и автоматизации проектирования АСОИУ.

Реализация модульного принципа проектирования АСОИУ предполагает, что каждый модуль обладает следующими качествами:

- функциональность, т.е. модуль должен представлять собой функционально законченную максимально независимую совокупность операций по обработке данных; обращение к модулю осуществляется как к единому целому, и значения вызываемых параметров обычно отражает специфику функций модуля;
- связность, т.е. модуль реализует совокупность взаимосвязанных функций, требующих одних и тех же данных; часть этих данных обычно скрыта для системы в целом;
- алгоритмичность, т.е. функции модуля группируются на алгоритмической основе;
- последовательность, т.е. модуль включает несколько функций, которые реализуются последовательно, причем выходные результаты одной функции являются входными для другой и т.д.; кроме того, функции модуля обычно являются взаимосвязанными во времени;
- однородность, т.е. в модуле объединяются однородные по своему функциональному назначению процедуры.

Основой для формализованной постановки и решения задач анализа и синтеза модульных АСОИУ является определение модулей системы и межмодульного интерфейса.

Могут быть выделены различные способы разбиения информационного и программного обеспечения АСОИУ на отдельные модули:

функциональный – по числу информационных и управляющих связей между модулями;

ресурсный – по имеющимся возможностям технического и программного обеспечения разработки;

элементный – по использованию типовых и стандартных элементов решений;

смешанный – обеспечивающий выше перечисленные.

Перейдем к формализации.

Пусть $A = \{a_1, a_2, \dots, a_m\}$ – множество задач, выявленных на предпроектной стадии. Каждое a_i в общем случае характеризуется n – мерным вектором показателей x_i , которыми являются время выполнения, число входных и выходных переменных, требуемый объем памяти и т.д.

Все задачи информационно взаимосвязаны. Это можно задать орграфом $\Gamma = (A, D)$, у которого вершины $A = \{a_1, a_2, \dots, a_m\}$ – это задачи, а ребра D – информационные связи между задачами (процедурами).

Пусть граф Γ задан матрицей смежности $\|d_{ij}\|$, причем $d_{ij} = 1$, если существует информационная дуга из задачи a_i в задачу a_j , и $d_{ij} = 0$ в противном случае. Каждая дуга между элементами a_i и a_j характеризуется некоторым параметром ρ_{ij} , который может быть и вектором. Будем считать, что $\rho_{ij} = 0$, если $d_{ij} = 0$.

Обозначим через $E = \{\theta\}$ – множество всех возможных разбиений множества A на отдельные подмножества, т.е. каждое θ таково, что

$$\Theta = (A_1^\theta, \dots, A_\ell^\theta, \dots, A_{L(\theta)}^\theta), \bigcup_{\ell=1}^{L(\theta)} A_\ell^\theta = A, A_i^\theta \cap A_j^\theta = \emptyset, i, j = 1, \dots, L(\theta), i \neq j.$$

Рассмотрим некоторое разбиение θ . Подмножество $A_\ell^\theta \in \theta$ будем в дальнейшем называть модулем.

Для данного разбиения θ множество дуг исходного графа Γ распадается на $L(\theta)+1$ попарно не пересекающихся подмножеств: а) подмножество внутреннее G_ℓ^θ дуг, соединяющих вершины из A_ℓ^θ ; б) подмножество внешнее D^θ дуг, концы которых лежат в разных модулях. Внешние дуги, входящие в какой-либо элемент модуля A_ℓ^θ называют его входом, а выходящие из какого-либо элемента – выходом.

Разбиению θ можно сопоставить агрегированный граф Γ^θ , получающийся из исходного графа Γ в результате объединения всех вершин подмножества A_ℓ^θ в одну для каждого $\ell = 1, \dots, L(\theta)$. Из вершины A_r^θ в вершину A_k^θ идет дуга тогда и только тогда, когда в графе Γ имеется дуга, направленная от некоторой вершины $a_r \in A_r^\theta$ в вершину $a_k \in A_k^\theta$. Дугам графа Γ^θ

$$\text{сопоставим параметры } q_{rk}^\theta = \sum_{(i,j) \in C_{rk}^\theta} \rho_{ij}, r \neq k,$$

$$\text{где } C_{rk}^\theta = \{(i,j): a_r \in A_r^\theta, a_k \in A_k^\theta, d_{ij} = 1\}.$$

Агрегированный граф Γ^θ описывает разбиение исходной системы на модули.

В зависимости от интерпретации дуг и вершин рассмотренной модели, используемых критериев оптимизации и ограничений могут быть рассмотрены различные способы разбиения графа Γ .

2. Постановка и формализация задачи разбиения ИЛИМ проектируемой АСОИУ на модули обработки данных с минимальным числом информационных связей между ними.

Данная задача возникает на этапе технического проектирования, в процессе которого формируются общие требования к системе, определяются выполняемые системой функции или процедуры по обработке входной информации

Исходными данными для задачи являются:

- 1) множество различных типов входных, промежуточных и выходных данных,
- 2) множество необходимых процедур преобразования данных.

При этом задачи относятся к одному и тому же уровню (см. л. 8).

Отношение, т.е. оператор отображения множества процедур к множеству информационных элементов можно представить в виде двудольного графа, дуги которого соединяют процедуры с соответствующими информационными данными (см. рис. 4.5.1).

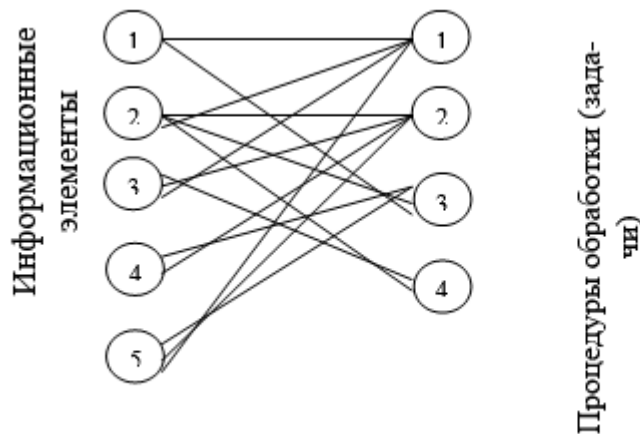


Рис. 4.5.1. Двудольный граф связи процедур и информационных элементов

Разбиение информационного и программного обеспечения АСУ на модули сводится в данной задаче к разбиению заданного множества задач (процедур обработки данных) на непересекающиеся подмножества, имеющие минимальное число общих информационных элементов.

Наиболее наглядно формализованная постановка задачи иллюстрируется с использованием графа, вершинами которого являются процедуры и информационные элементы, а связывающие их дуги соответствуют имеющимся информационным элементам.

Для формализации задачи введем следующие обозначения:

$A = \{a_1, a_2, \dots, a_j, \dots, a_m\}$ – множество задач, далее процедур системы обработки данных,

$R = \{r_1, r_2, \dots, r_l, \dots, r_n\}$ – множество информационных элементов,

$$d_{j\ell} = \begin{cases} 1, & \text{если информационный элемент } r_\ell \text{ используется для выполнения} \\ & \text{процедуры } a_j \\ 0, & \text{в противном случае} \end{cases}$$

Введем в рассмотрение следующие переменные:

$$x_{ij} = \begin{cases} 1, & \text{если процедура } a_j \text{ входит в модуль } A_i, j=1, \dots, m; i=1, \dots, M=2^m-1 \\ 0, & \text{в противном случае} \end{cases}$$

и связанные с ней переменные:

$$y_{\ell i} = \begin{cases} 1, & \text{если } \sum_{j=1}^m d_{j\ell} \cdot x_{ij} \geq 1; i=1, \dots, M; \ell=1, \dots, L \\ 0, & \text{если } \sum_{j=1}^m d_{j\ell} \cdot x_{ij} = 0 \end{cases}$$

т.е. $y_{\ell i}=1$, если r_{ℓ} -й информационный элемент используется процедурой a_i , которая размещается в модуле A_i .

При этих обозначениях суммарное число различных информационных элементов, являющихся общими для различных модулей $A_i, i=1, \dots, M$ равно

$$Z = \sum_{\ell=1}^L \sum_{i=1}^M \sum_{i'=i+1}^M y_{\ell i} \cdot y_{\ell i'} \rightarrow \min \quad (1)$$

При ограничениях:

$$1. \text{ На число выделяемых модулей } M_0 \leq M \quad (2)$$

Каждая процедура хотя бы в одном модуле должна находиться:

$$\sum_{i=1}^{M_0} x_{ij} \geq 1, j=1, \dots, m \quad (3)$$

$$2. \text{ Число процедур в одном модуле может быть ограничено некоторой величиной } N: \sum_{j=1}^m x_{ij} \leq N, i=1, \dots, M_0 \quad (4)$$

$$3. \text{ Некоторые процедуры } j \text{ и } j' \text{ должны быть разнесены по разным модулям, т.е.} \\ x_{ij} + x_{ij'} \leq 1, i=1, \dots, M_0 \quad (5)$$

$$4. \text{ Число информационных элементов, с которыми связан модуль, может быть ограничено, т.е. } \sum_{\ell=1}^L y_{\ell i} \leq k, \\ i=1, \dots, M_0 \quad (6)$$

3. Ограничения на сложность взаимодействия информационных элементов и процедур внутри модулей, т.е.

$$\sum_{\ell=1}^L \sum_{j=1}^m d_{j\ell} \cdot x_{ij} \leq G, i=1, \dots, M_0 \quad (7)$$

6. Ограничения на количество сложных связей между информационными элементами и некоторой парой i и i' модулей, т.е.

$$\sum_{\ell=1}^L y_{\ell i} \cdot y_{\ell i'} \leq R \quad (8)$$

Задача (1)-(8) является задачей квадратичного целочисленного программирования.

Для удобства решения она может быть сведена к линейной форме путем линеаризации выражений (1) и (8).

Введем переменную

$$Z_{\ell ii'} = \begin{cases} 1, & \text{если } \ell\text{-й информационный элемент необходим для модулей } A_i \text{ и } A_{i'} \\ 0, & \text{в противном случае} \end{cases}$$

Тогда критерий (1) может быть записан так:

$$\sum_{\ell=1}^L \sum_{i=1}^M \sum_{i'=1}^M Z_{\ell ii'} \rightarrow \min \quad (9)$$

Ограничение (8) при этом будет иметь вид:

$$\sum_{\ell=1}^m Z_{\ell ii'} \leq R \text{ для заданных } i \text{ и } i' \quad (10)$$

Задача разбиения, представленная выражениями (9), (2)-(7), (10) имеет линейный вид и может быть решена с использованием стандартных методов. Рассмотрим следующий пример решения задачи разбиения программного и информационного обеспечения АСОИУ на функциональные модули, имеющие минимальное число информационных связей.

Пусть граф задачи содержит пять процедур и шесть информационных элементов (см. рис. 4.5.2).

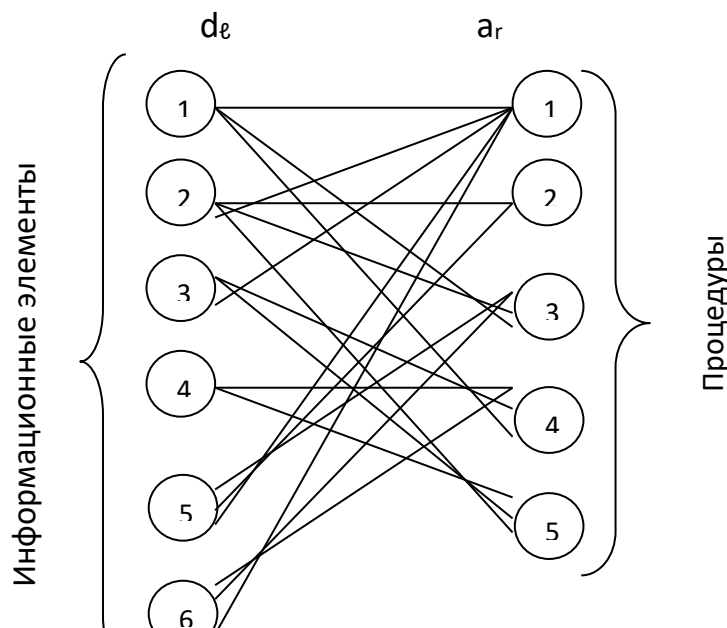


Рис. 4.5.2. Граф взаимосвязи процедур с информационными элементами

Матричная форма соотношений между процедурами и информационными элементами приведена в таблице 4.5.1.

Необходимо разбить множество процедур обработки данных на два модуля, имеющих минимальное число общих информационных элементов, причем разбиение необходимо провести таким образом, чтобы модули включали только соседние процедуры.

Рассмотрим решение задачи при следующих ограничениях:

- 1) общее число модулей $V=2$;
- 2) общее число информационных элементов в каждом из модулей $L_v \leq 6, v=1,2$;
- 3) общее число процедур в каждом модуле $M_v \leq 4, v=1,2$.

Таблица 4.5.1. Матричная форма соотношений

A _r	d _e					
	1	2	3	4	5	6
1	+	+	+		+	+
2		+			+	
3	+	+			+	+
4	+		+	+		
5		+	+	+		

Находим матрицу взаимосвязи процедур обработки с информационными элементами:

$$\|d_{\ell}\| = \begin{bmatrix} 111011 \\ 010010 \\ 110011 \\ 101101 \\ 011100 \end{bmatrix}$$

При заданных условиях возможно четыре варианта разбиения процедур на модули

I [a₁], [a₂, a₃, a₄, a₅]

II [a₁, a₂], [a₃, a₄, a₅]

III [a₁, a₂, a₃], [a₄, a₅]

IV [a₁, a₂, a₃, a₄], [a₅]

Для каждого варианта разбиения определяем матрицы X_{ji} и Y_{li} и значение критерия S:

$$\text{I } X_{ji}^T = \begin{bmatrix} 10 \\ 01 \\ 01 \\ 01 \\ 01 \end{bmatrix}; \quad Y_{li} = \begin{bmatrix} 11 \\ 11 \\ 11 \\ 01 \\ 11 \\ 11 \end{bmatrix}; \quad S_{\text{I}}=5$$

$$\text{II } X_{ji}^T = \begin{bmatrix} 10 \\ 10 \\ 01 \\ 01 \\ 01 \end{bmatrix}; \quad Y_{li} = \begin{bmatrix} 11 \\ 11 \\ 11 \\ 01 \\ 11 \\ 11 \end{bmatrix}; \quad S_{\text{II}}=5$$

$$\text{III } X_{ji}^T = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 01 \\ 01 \end{bmatrix}; \quad Y_{li} = \begin{bmatrix} 11 \\ 11 \\ 11 \\ 01 \\ 10 \\ 10 \end{bmatrix}; \quad S_{\text{III}}=3$$

$$\text{IV } X_{ji}^T = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 01 \end{bmatrix}; \quad Y_{li} = \begin{bmatrix} 10 \\ 11 \\ 11 \\ 11 \\ 10 \\ 10 \end{bmatrix}; \quad S_{\text{IV}}=3$$

По критерию $S = \sum_{l=1}^L \sum_{i=1}^V \sum_{i'=i+1}^V Y_{li} \cdot Y_{li'} \rightarrow \min$ оптимальным является IV вариант разбиения.

3. Постановка и модель решения задачи разбиения ИЛМ АСОИУ на функциональные модули с минимальным временем обмена с внешней памятью ЭВМ (базой данных)

В отличие от предыдущей постановки задачи разбиения ИЛМ АСОИУ на функциональные модули, в данном случае мы будем учитывать временные характеристики обработки информации в системе.

Кроме этого будем учитывать тот факт, что информационные элементы могут быть размещены в том или ином информационном массиве, к которому следует обращаться соответствующим процедурам в процессе обработки информации, причем эти процедуры размещены в тех или иных, заранее неизвестных модулях, число которых заранее определено (см. предыдущую задачу).

Определим вначале те исходные данные, которые являются известными к началу постановки и дальнейшей формализации поставленной задачи.

Предполагается известными количество функциональных модулей, на которые разбивается множество процедур обработки данных V . Но какие процедуры и в каком количестве они размещены в том или ином модуле неизвестно.

Кроме того, известно количество информационных массивов F , в которых размещены информационные элементы, но в каких массивах и в каком количестве их находится в том или ином массиве неизвестно.

Известными являются также следующие данные:

- $A = \{a_j; j=1, \dots, m\}$ – множество последовательно выполняемых процедур в проектируемой системе обработки данных

- $R = \{r_l; l=1, \dots, L\}$ – множество информационных элементов, обрабатываемых процедурами множества A

- $W^c = \|w_{jl}^c\|$ – матрица взаимосвязей информационных элементов с соответствующими процедурами обработки данных при считывании этих элементов из соответствующих массивов. Элемент этой матрицы w_{jl}^c определен следующим образом:

$$w_{jl}^c = \begin{cases} 1, & \text{если информационный элемент } l \text{ считывается из массива процедурой } j, \\ 0, & \text{если информационный элемент } l \text{ не связан с процедурой } j. \end{cases}$$

- $W^3 = \|w_{jl}^3\|$ – матрицы взаимосвязей информационных элементов с соответствующими процедурами обработки данных при их записи в соответствующий массив. Элемент этой матрицы w_{jl}^3 определен следующим образом:

$$w_{jl}^3 = \begin{cases} 1, & \text{если информационный элемент } l \text{ записывается в массив процедурой } j, \\ 0, & \text{если информационный элемент } l \text{ не связан с процедурой } j. \end{cases}$$

- τ_i – среднее время считывания i -го модуля из внешней памяти в оперативную память ЭВМ, $i = 1, \dots, V$;

- t_f^c – среднее время считывания f -го массива из внешней памяти в оперативную память ЭВМ, $f = 1, \dots, F$;

- t_f^3 – среднее время записи результатов в f -й массив, находящийся в оперативной памяти ЭВМ, $f = 1, \dots, F$.

Введем теперь следующие переменные, нахождение значений которых и будет результатом решения поставленной задачи.

$$x_{ij} = \begin{cases} 1, & \text{если } j\text{-я по порядку выполнения процедура включается в состав } i\text{-го модуля;} \\ 0, & \text{в противном случае;} \end{cases}$$

$i=1, \dots, V$; V – возможное число модулей; $j = 1, \dots, m$.

$$z_{lf} = \begin{cases} 1, & \text{если } l\text{-й информационный элемент включается в } f\text{-й массив;} \\ 0, & \text{в противном случае;} \end{cases}$$

$$f=1,\dots,F; F \leq L; l = 1, \dots, L.$$

Для удобства получения выражения для целевой функции данной постановки задачи введем еще переменные, которые являются связующими элементами между модулями и информационными элементами.

$$y_{il}^c = \begin{cases} 1, & \sum_{j=1}^m w_{jl}^c \times x_{ij} \geq 1; i = 1, \dots, V; j = 1, \dots, m; \\ 0, & \sum_{j=1}^m w_{jl}^c \times x_{ij} = 0; i = 1, \dots, V; j = 1, \dots, m. \end{cases}$$

$$y_{il}^3 = \begin{cases} 1, & \sum_{j=1}^m w_{jl}^3 \times x_{ij} \geq 1; i = 1, \dots, V; j = 1, \dots, m; \\ 0, & \sum_{j=1}^m w_{jl}^3 \times x_{ij} = 0; i = 1, \dots, V; j = 1, \dots, m. \end{cases}$$

Другими словами, $y_{il}^c(y_{il}^3) = 1$, если при выполнении i -го модуля, т.е. входящих в него процедур обработки данных, требуется информационный элемент с номером l , который или считывается из соответствующего массива или в него записывается.

Для того, чтобы связать массивы, в которых содержатся соответствующие информационные элементы с теми процедурами, которые расположены в соответствующих модулях, и которые требуются для этих процедур, введем еще две переменных z_{if}^c и z_{if}^3 , которые определим следующим образом:

$$z_{if}^c = \begin{cases} 1, & \sum_{l=1}^L y_{il}^c \times z_{lf} \geq 1; \\ 0, & \sum_{l=1}^L y_{il}^c \times z_{lf} = 0. \end{cases}$$

$$z_{if}^3 = \begin{cases} 1, & \sum_{l=1}^L y_{il}^3 \times z_{lf} \geq 1; \\ 0, & \sum_{l=1}^L y_{il}^3 \times z_{lf} = 0. \end{cases}$$

Другими словами, $z_{if}^c(z_{if}^3) = 1$, если массив с номером f содержит хотя бы один информационный элемент, необходимый для выполнения хотя бы одной процедуры из модуля с номером i .

Таким образом, переменные $y_{il}^{c(3)}$ и $z_{if}^{c(3)}$ служат для формализации взаимосвязи системы разрабатываемых модулей с отдельными информационными элементами и массивами при считывании и записи в процессе обмена с внешней памятью ЭВМ.

Задачу разбиения ИЛМ АСОИУ на функциональные модули с минимальным временем обмена с внешней памятью ЭВМ (базой данных) можно теперь формализовать формулируется следующим образом.

Запишем вначале критерий оптимизации, т.е. время обработки информации в системе, которое минимизируется. Оно складывается из времени считывания модулей, где содержатся соответствующие процедуры обработки данных, и времени вызова соответствующих массивов, где содержатся соответствующие информационные элементы. Этот критерий, с учетом ранее введенных обозначений, можно записать следующим образом:

$$\left\{ \sum_{i=1}^V \left(\left(\sum_{j=1}^m x_{ij} (1 - x_{i,j+1}) \right) \times \tau_i + \sum_{f=1}^F z_{if}^c \times t_f^c + \sum_{f=1}^F z_{if}^3 \times t_f^3 \right) \right\} \rightarrow \min_{\{x_{if}, z_{if}\}}$$

В этом выражении 1-е слагаемое – условие обработки информации всеми процедурами модуля, причем одновременно 2 модуля не обрабатываются и учет времени считывания модулей, причем процедуры выполняется последовательно; 2-е и 3-е слагаемые – учет времени считывания (записи) информационных массивов.

Рассмотрим теперь систему ограничений, без которой критерий не имеет смысла, так как он принимает минимальное значение, если всем переменным присвоить значения, равные нулю.

Таковыми ограничениями являются следующие ограничения, которые во многом определяются или содержательным анализом проектируемой системы, или возможностями технических средств, на которые будут возложена задача обработки данных в системе:

- на общее число процедур в составе каждого модуля:

$$\sum_{j=1}^m x_{ij} \leq \overline{M}_i, \quad i = \overline{1, V}, \text{ где } \overline{M}_i - \text{допустимое число процедур в } i\text{-ом модуле};$$

- на число информационных элементов, обрабатываемых процедурами каждого модуля:

$$\sum_{\ell=1}^L y_{i\ell} \leq \overline{L}_i, \quad i = \overline{1, V},$$

где \overline{L}_i – максимальное допустимое число информационных элементов, обрабатываемых i -ым модулем;

- на сложность интерфейса между всеми модулями системы обработки данных:

$$\sum_{\ell=1}^L \sum_{i=1}^V \sum_{i'=i+1}^V y_{i\ell} y_{i'\ell} \leq \overline{S},$$

где \overline{S} – максимально допустимый межмодульный интерфейс между модулями СОД, т.е. допустимое число переменных, информационных элементов, являющихся общими для выделенных модулей;

- на сложность интерфейса между отдельными модулями системы обработки данных:

$$\sum_{\ell=1}^L y_{i\ell} \cdot y_{i'\ell} \leq \overline{S}_{ii'}, \text{ для заданных } i \text{ и } i',$$

где $\overline{S}_{ii'}$ – максимальное число общих переменных (информационных элементов), обрабатываемых модулями i и i' ;

- на однократность включения процедур в программные модули:

$$\sum_{i=1}^V x_{ij} = 1, \quad j = \overline{1, m};$$

- на включение отдельных процедур в состав одного модуля:

$$x_{ij} + x_{i'j} \leq 1, \text{ для заданных } j \text{ и } j', i = \overline{1, V};$$

- на передачу управления из модуля до завершения обработки информации всеми процедурами модуля:

$$\sum_{j=1}^m x_{ij} \cdot (1 - x_{ij+1}) = 1, \quad i = \overline{1, V};$$

– на дублирование информационных элементов в массивах:

$$\sum_{f=1}^F z_{\ell f} = k, \quad \ell = \overline{1, L}; \quad k = \overline{1, K}, \text{ где } k - \text{допустимая степень дублирования информационных элементов в массивах системы};$$

– на размер записи каждого массива:

$$\sum_{\ell=1}^L z_{\ell f} \leq \overline{N_f}; \quad f = \overline{1, F}, \text{ где } \overline{N_f} - \text{максимально допустимое число информационных элементов в } f\text{-ом массиве.}$$

В заключение заметим, что данная задача относится к классу нелинейных задач математического программирования с булевыми переменными.

4. Постановка общей задачи синтеза информационного обеспечения АСОИУ модульного типа. Формализация задачи определения числа и состава информационных массивов.

Модульное построение проектируемой АСОИУ накладывает ряд условий на синтез информационного обеспечения. Основными из них являются необходимость учета динамики реализации модулей и вызова в оперативную память ЭВМ соответствующих информационных массивов в целом либо их частей.

Синтез информационного обеспечения проектируемой по модульному принципу АСОИУ может быть осуществлен в три этапа.

На первом этапе определяются число и состав информационных массивов при заданном составе и частоте использования каждого модуля. Критерием оптимизации при решении этой задачи является минимизация общего числа обменов проектируемой системы с внешней памятью ЭВМ.

На втором этапе решается задача выбора оптимальных методов организации данных в полученной на первом этапе системе информационных массивов и их размещение во внешней памяти ЭВМ. Критериями решения этих задач могут быть

- 1) минимизация времени обмена с внешней памятью,
- 2) минимизация суммарных затрат на хранение и использование полученной системы информационных массивов,
- 3) минимизация времени решения одной какой-либо, обычно наиболее важной, задачи.

На третьем этапе решается задача определения оптимальной величины блока данных, при которой минимизируется общее время считывания, поиска и передачи данных в оперативную память.

Схема постановки и последовательности решения задач синтеза ИО АСОИУ модульного типа имеет вид, представленный на следующем рисунке (см. рис. 5.4.1)

Покажем, какие оптимизационные модели могут быть использованы на каждом из этих этапов

Задача и модель определения числа и состава информационных массивов

Пусть в результате диагностического анализа проектируемой АС для заданного комплекса задач управления определено множество программных модулей $M = \{m_1, m_2, \dots, m_v, \dots, m_V\}$, где V – число таких модулей. Для каждого модуля m_v установлена средняя частота его функционирования $Q_v (v = \overline{1, V})$ в заданный интервал времени, например, в сутки. Известно также множество информационных элементов $D = \{d_1, d_2, \dots, d_\ell, \dots, d_L\}$, где L – число таких элементов, с которыми связаны модули множества M .

Каждый информационный элемент d_ℓ характеризуется длиной записи, например, в байтах λ_ℓ . Величины $\lambda_\ell (\ell = \overline{1, L})$ образуют вектор $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_\ell, \dots, \lambda_L\}$.

Для каждого модуля m_v ($v=1,...,V$) информационный элемент d_ℓ ($\ell=1,...,L$) может быть входом, выходом или модуль m_v никак не связан с информационным элементом d_ℓ . В первом случае будем говорить, что модуль m_v считывает элемент d_ℓ , а во втором – модуль m_v записывает элемент d_ℓ .

Связь между модулями и информационными элементами может быть задана в виде двух матриц, $B^c = \|b_{v\ell}^c\|$ и $B^3 = \|b_{v\ell}^3\|$, где $b_{v\ell}^c$ ($b_{v\ell}^3$) равен единице, если ℓ -й информационный элемент ($\ell=1,...,L$) считывается (записывается) v -м модулем ($v=1,...,V$) и равен нулю в противном случае.

Обозначим через F возможное число информационных массивов, по которым распределяются информационные элементы. Очевидно, что $F \leq L$.

Введем следующие булевы переменные:

$$x_{\ell f} = \begin{cases} 1, & \text{если } \ell\text{-й информационный элемент размещен в } f\text{-й массив} \\ 0, & \text{в противном случае.} \end{cases}$$

$$z_{vf}^c = \begin{cases} 1, & \text{если } \sum_{\ell=1}^L b_{v\ell}^c \cdot x_{\ell f} \geq 1; \\ 0, & \text{если } \sum_{\ell=1}^L b_{v\ell}^c \cdot x_{\ell f} = 0; \end{cases} \quad (1)$$

$$z_{vf}^3 = \begin{cases} 1, & \text{если } \sum_{\ell=1}^L b_{v\ell}^3 \cdot x_{\ell f} \geq 1; \\ 0, & \text{если } \sum_{\ell=1}^L b_{v\ell}^3 \cdot x_{\ell f} = 0. \end{cases} \quad (2)$$

Другими словами $z_{vf}^{c(3)}$ принимает значение 1, если модуль m_v связан с информационным элементом d_ℓ , который находится в массиве f .

Переменные $x_{\ell f}$ ($\ell=1,...,L$; $f=1,...,F$) образуют матрицу $X = \|x_{\ell f}\|$, определяющую распределение информационных элементов по информационным массивам. Матрицы $Z^c = \|z_{vf}^c\|$ и $Z^3 = \|z_{vf}^3\|$ размерности $V \times F$ каждая определяют связь программных модулей с информационными массивами.

Сформулированную задачу определения числа и состава информационных массивов можно теперь формально представить в виде следующей модели:

$$R = \sum_{v=1}^V Q_v \cdot \sum_{f=1}^F (z_{vf}^c + z_{vf}^3) \rightarrow \min. \quad (3)$$

при ограничениях

$$\sum_{\ell=1}^L x_{\ell f} \leq N_f \quad (f = \overline{1, F}) \quad (\text{общее число информационных элементов в массиве}) \quad (4)$$

$$\sum_{\ell=1}^L \lambda_\ell \cdot x_{\ell f} \leq \tilde{N}_f \quad (f = \overline{1, F}) \quad (\text{длина записи в массиве}) \quad (5)$$

$$\sum_{f=1}^F x_{\ell f} = 1 \quad (\ell = \overline{1, L}) \quad (\text{дублирование элементов в разных массивах}) \quad (6)$$

где z_{vf}^c и z_{vf}^3 определяются выражениями (1) и (2);

N_f – ограничения на общее число информационных элементов в f -м массиве;

\tilde{N}_f – ограничения на длину записи в f -м массиве.

Значения величин \tilde{N}_f , N_f , $f=1, \dots, F$ определяются разработчиком АСУ исходя из ограничений, наложенных на внешние запоминающие устройства выбранной ЭВМ.

Задача (1) – (6) относится к классу задач целочисленного нелинейного программирования и может быть решена методом ветвей и границ. Алгоритм решения этой задачи будет рассмотрен в этом семестре в лабораторной работе.

Данной задаче можно придать графовую интерпретацию (см. рис. 5.4.2).

Необходимо синтезировать двудольный граф $\tilde{G} = (M, B, Z^{(3)})$, имеющий минимальное число дуг при ограничениях на размеры и сложность информационных массивов, возможности дублирования и распределения информационных элементов по массивам, где:

M – множество модулей: $M = \{m_v; v=1, \dots, V\}$;

B – множество определяемых информационных массивов: $B = \{b_f; f=1, \dots, F\}$;

$Z^{(3)}$ – множество дуг, связывающих множество модулей с множеством массивов: $Z^{(3)} = \parallel Z_M^{(3)} \parallel$.

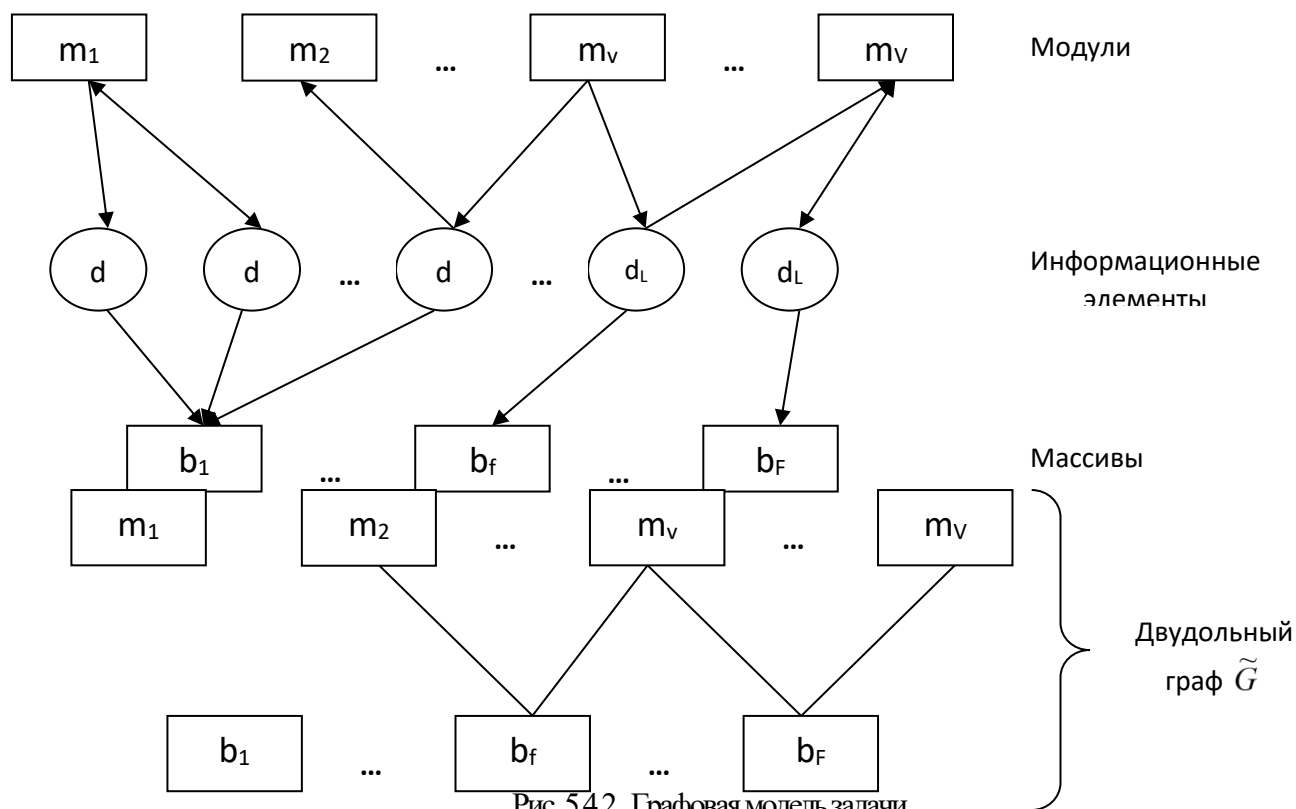


Рис. 5.4.2. Графовая модель задачи

Рассмотрим следующий пример. Пусть после исследования проектируемой АСУ было выяснено, что необходимы 3 модуля, связанные с 6 информационными элементами. Эта связь и частота использования модулей в процессе функционирования АСОИУ приведены в таблице 5.4.1.

Номер модуля	Информационные элементы, считываемые модулями из БД	Информационные элементы, записываемые модулями в БД	Средняя частота использования модуля в единицу времени
1	2, 3, 4, 5	1, 4	10
2	1, 2, 6	5	12
3	4, 5, 6	3	8

Длины соответствующих информационных элементов в килобайтах приведены в таблице 5.4.2.

Таблица 5.4.2

Информационный элемент	1	2	3	4	5	6
Длина элемента, (КБ)	20	30	20	20	40	50

Представим исходные данные в виде направленного двудольного графа $G = G(M, D, B)$, где $M = \{m_1, m_2, m_3\}$ - множество модулей системы, $D = \{d_1, d_2, d_3, d_4, d_5, d_6\}$ - множество информационных элементов, $B = B^c + B^s$ - матрица смежности графа, показывающая связи модулей с информационными элементами и имеющая следующий вид (см. табл. 1):

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ Граф } G = G(M, D, B)$$

изображен на рис. 1, где m_1, m_2, m_3 - модули; d_1, \dots, d_6 - информационные элементы; a_1, a_2, a_3 - массивы. Дуга направлена от модуля к элементу, если этот элемент записывается модулем, и от элемента к модулю, если элемент считывается модулем.

Один из вариантов построения двудольного графа G' изображен на рис. 2. Он получен при условии, что информационные элементы $d_1, d_2, d_3, d_4, d_5, d_6$ размещены в массивах a_1, a_2, a_3 так, как показано на рис. 5.4.3.

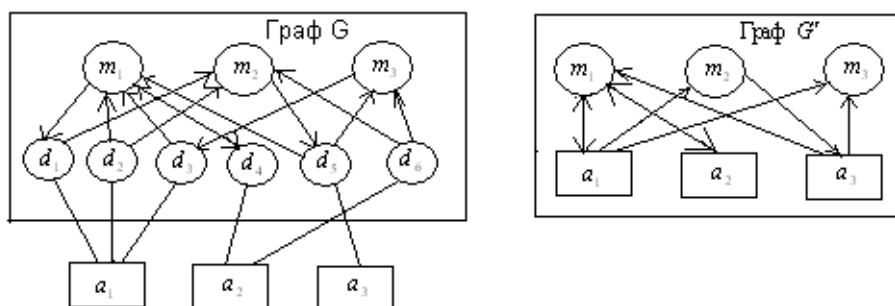


Рис. 1

Рис. 5.4.3.

Среднее число обменов системы программных модулей с внешней памятью (базой данных) при размещении информационных элементов по массивам так, как показано на рис. 1, равно 100. Если, однако, в массив a_1 поместить элементы d_1, d_2, d_3 , в массив a_2 - элементы d_4, d_5 , а в массив a_3 - элемент d_6 , то среднее число обменов сократится до 90. Таким образом, имеет место улучшение ранее составленного варианта размещения элементов по массивам.

Рассматриваемая задача относится к классу задач нелинейного программирования с булевыми переменными.

Если число информационных элементов L относительно невелико, то ее решение можно получить методом прямого перебора всех вариантов распределения информационных элементов по

информационным массивам. Для этого строят дерево вариантов по следующей схеме: первый информационный элемент помещают в первый информационный массив; второй – в первый массив или во второй; третий – в первый, второй или третий массив и т.д.

Очевидно, что число слоев такого дерева равно L , а число вершин в k -ом слое (k - номер информационного элемента) равно $k \cdot (k - 1)$.

После построения L -го слоя дерева вариантов для каждого из них проверяются условия (4), (5) и (6) и среди им удовлетворяющих выбирается вариант с минимальным значением функционала (3).

При большом числе информационных элементов L более рациональным является использование метода «ветвей и границ».

Точная нижняя оценка множества решений R_H достигается в том случае, если все информационные элементы распределены в один информационный массив. Учитывая, что каждый считываемый или записываемый информационный элемент, по крайней мере, один раз должен быть использован хотя бы в одном программном модуле, то из выражения (3) непосредственно следует, что для вычисления R_H справедлива следующая формула:

$$R_H = 2 \sum_{v=1}^V Q_v. \quad (7)$$

Точная верхняя граница (оценка) множества решений R_B достигается, если для каждого информационного элемента выделен свой информационный массив. В этом случае из выражения (3) непосредственно следует, что для вычисления R_B справедлива следующая формула:

$$R_B = \sum_{l=1}^L \sum_{v=1}^V Q_v (b_{vl}^c + b_{vl}^3). \quad (8)$$

Оценку R_n любого подмножества решений в n -й вершине дерева ветвления можно определить по формуле:

$$R_n = R_H + \Delta R_n, \quad (9)$$

где ΔR_n - приращение оценки в n -й вершине дерева ветвления.

Величина оценки R_n любого подмножества решений в n -й вершине ветвления зависит от множества значений $\{x_{lf}\}$ переменных, зафиксированных в этой n -й вершине в процессе ветвления, и может быть вычислена по следующей формуле:

$$\Delta R_n = \sum_{v=1}^V Q_v (P_v^c + P_v^3). \quad (10)$$

Величины P_v^c и P_v^3 в выражении (10) определяются по следующим формулам:

$$P_v^c = \begin{cases} \sum_{f=1}^{F_n} z_{vf}^c - 1, & \sum_{f=1}^{F_n} z_{vf}^c > 1; \\ 0, & \sum_{f=1}^{F_n} z_{vf}^c \leq 1; \end{cases} \quad (11)$$

$$P_v^3 = \begin{cases} \sum_{f=1}^{F_n} z_{vf}^3 - 1, & \sum_{f=1}^{F_n} z_{vf}^3 > 1; \\ 0, & \sum_{f=1}^{F_n} z_{vf}^3 \leq 1. \end{cases} \quad (12)$$

В выражениях (11) и (12) величина F_n - это число образуемых массивов в n -й вершине дерева ветвления, а z_{vf}^c и z_{vf}^3 - булевы переменные, определяемые по формулам (1) и (2) соответственно при условии, что информационные элементы распределены по информационным массивам согласно n -й вершине дерева ветвления.

Алгоритм решения задачи определения числа и состава информационных массивов можно теперь представить в виде следующей последовательности шагов.

1. Вводят исходные данные задачи в виде массивов M , D , Λ и матриц B^c и B^3 .
2. По формулам (7) и (8) вычисляют верхнюю и нижнюю границы R_B и R_H .
3. Распределяют информационный элемент d_1 в информационный массив a_1 (строят первый слой дерева вариантов).
4. Полагая $i = 1, 2$, помещают элемент d_i последовательно в информационный массив с номером i (строят второй слой дерева вариантов).
5. Для каждой вершины k -го слоя (варианта распределения рассматриваемых информационных элементов по информационным массивам) последовательно проверяют выполнимость условий (4) и (6), полагая $F = F_n$, где n - номер рассматриваемой вершины. Для тех вершин, у которых условия (4) и (6) не выполняются, полагают $R_n = R_B$, для остальных вершин значения R_n вычисляют по формулам (9) – (12).
6. Находят вершину рассматриваемого k -го слоя с номером n^* таким, что имеет место равенство $R_{n^*} = \min_{n \in S_k} \{R_n\}$, где S_k - множество номеров построенных вершин этого слоя.
7. Проверяют, все ли слои дерева вариантов построены. Если да, то выполняют п. 9, в противном случае выполняют пункт 8.
8. Запоминают номер вершины n^* рассматриваемого k -го слоя и строят вершины $(k+1)$ -го слоя, взяв в качестве исходной вершину с номером n^* . После построения всех вершин $(k+1)$ -го слоя переходят к п. 5.
9. Находят решение задачи определения числа и состава информационных массивов, выбирая в каждом слое построенного дерева вариантов вершины с соответствующими номерами n^* .

Дерево вариантов применительно к рассмотренной выше задаче распределения 6 информационных элементов по информационным массивам приведены на рис. 5.4.4.

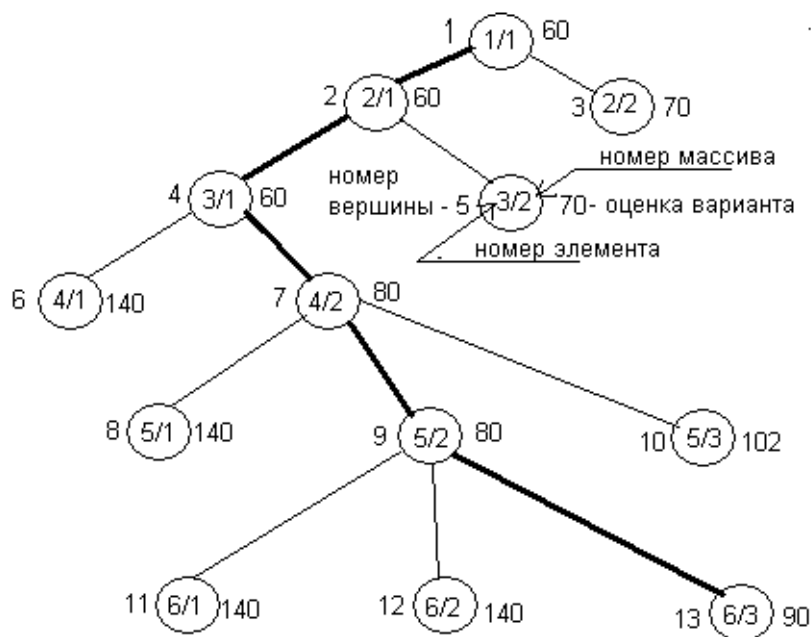


Рис. 2

Рис. 5.4.4.

Точное решение этой задачи состоит в следующем: 1-й, 2-й и 3-й информационные элементы помещают в 1-й информационный массив; 4-й и 5-й информационные элементы помещают во 2-й информационный массив; 6-й информационный элемент составляет 3-й информационный массив. Общее время считывания, поиска и передачи требуемых данных при этом равно 90.

5. Постановка общей задачи синтеза информационного обеспечения АСОИУ модульного типа. Формализация задачи выбора оптимальных методов организации массивов и размещения модулей и массивов во внешней памяти ЭВМ. Формализация задачи определения величины блока данных.

6. Постановка общей задачи синтеза информационного обеспечения АСОИУ модульного типа. Формализация задачи определения величины блока данных.

Модульное построение проектируемой АСОИУ накладывает ряд условий на синтез информационного обеспечения. Основными из них являются необходимость учета динамики реализации модулей и вызова в оперативную память ЭВМ соответствующих информационных массивов в целом либо их частей.

Синтез информационного обеспечения проектируемой по модульному принципу АСОИУ может быть осуществлен в три этапа.

На первом этапе определяются число и состав информационных массивов при заданном составе и частоте использования каждого модуля. Критерием оптимизации при решении этой задачи является минимизация общего числа обменов проектируемой системы с внешней памятью ЭВМ.

На втором этапе решается задача выбора оптимальных методов организации данных в полученной на первом этапе системе информационных массивов и их размещение во внешней памяти ЭВМ. Критериями решения этих задач могут быть

- 4) минимизация времени обмена с внешней памятью,
- 5) минимизация суммарных затрат на хранение и использование полученной системы информационных массивов,

б) минимизация времени решения одной какой-либо, обычно наиболее важной, задачи.

На третьем этапе решается задача определения оптимальной величины блока данных, при которой минимизируется общее время считывания, поиска и передачи данных в оперативную память.

Схема постановки и последовательности решения задач синтеза ИО АСОИУ модульного типа имеет вид, представленный на следующем рисунке (см. рис. 5.4.1)

Покажем, какие оптимизационные модели могут быть использованы на каждом из этих этапов.

Как об этом уже говорилось, критериями, используемыми для решения данных задач, являются: 1) минимизация суммарных затрат на создание, хранение и эксплуатацию информационных массивов и программных модулей СОД либо 2) минимизация общего времени обработки данных или времени решения одной из задач обработки данных.

Введем необходимые обозначения:

$I = \{1, 2, \dots, i, \dots, I_0\}$ – множество задач обработки данных;

$N = \|n_{iv}\|$ – матрица принадлежности модуля к задаче, т.е.

$$n_{iv} = \begin{cases} 1, & \text{если модуль } v \text{ используется для решения задачи } i; \\ 0 & \text{в противном случае.} \end{cases}$$

$M^{(3)} = \|m_{vf}^{(3)}\|$ – матрица принадлежности массива к модулю,

т.е.

$$m_{vf}^{(3)} = \begin{cases} 1, & \text{если } f\text{-й массив считывается (записывается) } v\text{-м модулем} \\ 0 & \text{в противном случае.} \end{cases}$$

P_i – частота решения i -й задачи в АСОИУ;

q_v^i – частота использования v -го модуля при решении i -й задачи;

N_f – количество информационных элементов в одной записи f -го массива;

L_f – число записей в f -ом массиве;

$R_f = N_f * L_f$ – объем (размер) информационного массива f (общее число информационных элементов в массиве);

C_0 – стоимость единицы рабочего времени процессора для решения вычислительных задач;

C_v – приведенная стоимость единицы рабочего времени носителя информации v -го типа с внешней памятью ЭВМ;

S_v – стоимость блока управления v -го типа носителя информации; $v=1, \dots, v_0$

τ_v^0 – время считывания v -го модуля, размещенного на v -м типе носителя;

$t_{fi}^{v(c)}, t_{fi}^{v(3)}$ – время считывания (записи) f -го массива, организованного с использованием μ -го способа (можно

использовать разные способы доступа к данным: прямой, произвольный по ключам и т.п., можно по-разному организовывать саму структуру данных: реляционная, иерархическая, сетевая и смешанная) и размещенного на v -м типе носителя информации;

T_v – процессорное время реализации v -го модуля;

d_v – объем запоминающего устройства v -го типа носителя информации;

a_v – размер v -го модуля;

$\Delta \tau_v$ – время работы процессора при поиске v -го модуля;

$\Delta \tau_f^c (\Delta \tau_f^3)$ – время работы процессора при считывании (записи) f -го массива.

Введем переменные:

$$x_{fi}^{v\mu} = \begin{cases} 1, & \text{если } f\text{-й массив организован } \mu\text{-м методом и размещен на } v\text{-м типе носителя информации} \\ 0 & \text{в противном случае.} \end{cases}$$

$$y_v^v = \begin{cases} 1, & \text{если } v\text{-й модуль размещен на } v\text{-м типе носителя информации} \\ 0 & \text{в противном случае.} \end{cases}$$

1) Рассмотрим вначале постановку задачи выбора оптимальных методов организации информационных массивов, размещения массивов и модулей во внешней памяти, минимизирующих суммарные затраты на создание, хранение и эксплуатацию модульной АСОИУ.

Полные приведенные затраты C на решение задач АСОИУ являются суммой капитальных C_k и эксплуатационных затрат C_3 , т.е. $C = C_k + C_3$.

Капитальные затраты C_k определяются выбором типа носителя информации, т.е. выбором технических средств и могут быть определены следующим выражением:

$$C_k = \sum_{v=1}^{\nu_0} S_v \cdot \left[\underbrace{\left(\sum_{v=1}^V a_v \cdot y_v^v + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} R_f \cdot x_{f\mu}^v \right)}_{\text{объем информации на } v\text{-м носителе}} \cdot d_v^{-1} \right] \quad (1)$$

$\lfloor \alpha \rfloor$ - наименьшая целая часть, большая или равная α , где α - число носителей памяти типа $v=1, \dots, \nu_0$.

Эксплуатационные затраты, в общем случае, содержат следующие составляющие:

- Стоимость непосредственной вычислительной работы процессора $C_3^{\text{об}}$; при решении задач $i=1, \dots, I_0$;
- Стоимость процессорного времени при формировании адресов $C_3^{\text{ФА}}$; для поиска нужных модулей и информационных элементов в соответствующих массивах;
- Стоимость записи и считывания массивов и модулей, т.е. стоимость обмена между оперативной и внешней памятью $C_3^{\text{обм}}$. *- важно только это слагаемое*

Для вычисления этих составляющих могут быть использованы следующие формулы:

$$C_3^{\text{обп}} = C_0 \cdot \sum_{i=1}^{I_0} P_i \cdot \sum_{v=1}^V n_{iv} \cdot q_v^i \cdot T_v = C_0 \cdot \sum_{v=1}^V T_v \cdot \sum_{i=1}^{I_0} P_i \cdot n_{iv} \cdot q_v^i = C_0 \cdot \sum_{v=1}^V T_v \cdot Q_v \quad (2)$$

$$C_3^{\text{обм}} = \sum_{v=1}^{\nu_0} C_v \cdot \left\{ \sum_{i=1}^{I_0} P_i \cdot \sum_{v=1}^V n_{iv} \cdot q_v^i \cdot \left[\tau_v^v \cdot y_v^v + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^v \cdot \left(m_{vf}^c \cdot t_{f\mu}^{v(c)} + m_{vf}^3 \cdot t_{f\mu}^{v(3)} \right) \right] \right\} =$$

$$= \sum_{v=1}^{\nu_0} C_v \cdot \left\{ \sum_{v=1}^V Q_v \cdot \left[\tau_v^v \cdot y_v^v + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^v \cdot \left(m_{vf}^c \cdot t_{f\mu}^{v(c)} + m_{vf}^3 \cdot t_{f\mu}^{v(3)} \right) \right] \right\} \quad (3)$$

$$C_3^{\text{ФА}} = C_0 \cdot \left[\sum_{i=1}^{I_0} P_i \cdot \sum_{v=1}^V n_{iv} \cdot q_v^i \cdot \Delta \tau_v + \sum_{i=1}^{I_0} P_i \cdot \sum_{v=1}^V n_{iv} \cdot q_v^i \cdot \sum_{f=1}^F \left(m_{vf}^c \cdot \Delta \tau_{vf}^c + m_{vf}^3 \cdot \Delta \tau_{vf}^3 \right) \right] \quad (4)$$

Так как выражения (2) и (4) не содержат введенных переменных, то задача выбора способов организации и размещения модулей и массивов во внешней памяти формализуется следующим образом:

$$\text{найти } \min_{\{y_v^v, x_{f\mu}^v\}} \{C_k + C_3^{\text{обм}}\},$$

где C_k и $C_3^{\text{обм}}$ определяются формулами (1) и (3), при следующих ограничениях:

– на время T_i обмена с внешней памятью ЭВМ при решении i -й задачи:

$$\sum_{v=1}^V n_{iv} \cdot P_i \cdot q_v^i \cdot \sum_{v=1}^{v_0} \left[\tau_v^\nu \cdot y_v^\nu + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^\nu \cdot (m_{vf}^c \cdot t_{f\mu}^{\nu(c)} + m_{vf}^z \cdot t_{f\mu}^{\nu(z)}) \right] \leq T_i \quad (5)$$

– на используемый объем носителя информации ν -го типа:

$$\sum_{v=1}^V a_v \cdot y_v^\nu + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} R_f \cdot x_{f\mu}^\nu \leq G_\nu, \quad \nu = \overline{1, v_0} \quad (6)$$

– на совместное размещение модулей и массивов в одном блоке носителя ν -го типа:

$$y_v^\nu + x_{f\mu}^\nu \leq 1 \text{ для заданных } \nu \text{ и } (f, \mu); \quad (7)$$

– на размещение модулей на различных носителях:

$$\sum_{v=1}^V y_v^\nu = 1, \quad \nu = \overline{1, V} \quad (8)$$

– на размещение массивов на различных носителях:

$$\sum_{\mu=1}^{\mu_0} \sum_{v=1}^{v_0} x_{f\mu}^\nu = 1, \quad f = \overline{1, F} \quad (9)$$

– на размещение модулей и массивов на носителях определенного типа:

$$\sum_{v=1}^{v_0} y_v^\nu = 1, \quad \text{для заданного } \nu \quad (10)$$

$$\sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^\nu = 1, \quad \text{для заданного } \nu. \quad (11)$$

2) Рассмотрим постановку и решение задачи выбора оптимальных методов организации массивов и модулей во

внешней памяти, минимизирующих: 1) общее время обработки данных; 2) время решения одной из задач управления.

В первом случае критерий имеет вид:

$$\min_{\{x_{f\mu}^\nu, y_v^\nu\}} \sum_{v=1}^V \sum_{v=1}^{v_0} Q_v \cdot \left[\tau_v^\nu \cdot y_v^\nu + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^\nu \cdot (m_{vf}^c \cdot t_{f\mu}^{\nu(c)} + m_{vf}^z \cdot t_{f\mu}^{\nu(z)}) \right] \quad (12)$$

Во втором – зафиксировав некоторое i :

$$\min_{\{x_{f\mu}^\nu, y_v^\nu\}} \sum_{v=1}^V n_{iv} \cdot P_i \cdot q_v^i \cdot \sum_{v=1}^{v_0} \left[\tau_v^\nu \cdot y_v^\nu + \sum_{f=1}^F \sum_{\mu=1}^{\mu_0} x_{f\mu}^\nu \cdot (m_{vf}^c \cdot t_{f\mu}^{\nu(c)} + m_{vf}^z \cdot t_{f\mu}^{\nu(z)}) \right] \quad (13)$$

В этих задачах кроме ограничений (5) – (11) учитывается дополнительное ограничение на суммарные затраты Π на создание и эксплуатацию АСУ:

$$C_k + C_z^{\text{обм}} \leq \Pi \quad (14),$$

где C_k и $C_z^{\text{обм}}$ определяются формулами (1) и (3).

Сформулированные задачи являются задачами целочисленного линейного программирования с булевыми переменными.

5.4.4. Задача определения оптимальной величины блока данных

В результате выше описанных этапов синтеза информационного обеспечения модульной АСУ получены следующие исходные данные, необходимые для определения оптимальной величины блока обмена данными с внешней памятью:

- система модулей программного обеспечения;
- множества информационных массивов и их связей с системой модулей;
- способы и характеристики размещения массивов и программных модулей на устройствах внешней памяти.

Введем необходимые переменные и обозначения:

L_{fv} – число записей f -го массива, размещенного на v -м типе запоминающего устройства;

$M = \|m_{vf}\|$, $v=1, \dots, V$; $f=1, \dots, F$ – матрица связи массивов с модулями системы:

$$m_{vf} = \begin{cases} 2, & \text{если } f\text{-й массив считывается и записывается } v\text{-м модулем,} \\ 1, & \text{если } f\text{-й массив только считывается } v\text{-м модулем,} \\ 0, & \text{если } f\text{-й массив не используется } v\text{-м модулем;} \end{cases}$$

X_{fv} – целочисленная переменная, определяющая число записей в блоке при считывании и записи в f -й массив, размещенный на v -м типе запоминающего устройства;

Требуется выбрать множество переменных $\{x_{fv}\}$ таким образом, чтобы минимизировать общее число обращений к внешней памяти с учетом технологических ограничений. Эта задача формулируется следующим образом:

$$\min_{\{x_{fv}\}} \sum_{f=1}^F \sum_{v=1}^{V_0} \sum_{v=1}^V]m_{vf} \cdot L_{fv} \cdot x_{fv}^{-1} [\quad (15)$$

при ограничениях:

- на объем оперативной памяти D_v , доступной для данного v -го модуля:

$$\sum_{f=1}^F \sum_{v=1}^{V_0} m_{vf} \cdot x_{fv} \leq D_v, \quad v = \overline{1, V} \quad (16)$$

- на допустимый минимальный \underline{d}_v и максимальный \overline{d}_v объема блока, размещенного в v -м типе запоминающего устройства:

$$\underline{d}_v \leq x_{fv} \leq \overline{d}_v, \quad f=1, \dots, F; v=1, \dots, V_0 \quad (17)$$

- на целочисленность величины блока:

$$x_{fv} \geq 1, \text{ где } x_{fv} - \text{целое, } f=1, \dots, F; v=1, \dots, V_0 \quad (18)$$

Данная задача является задачей дискретного программирования с нелинейной целевой функцией и линейными ограничениями и может быть решена алгоритмом, основанным на схеме «ветвей и границ».

7. Основные показатели надежности АСОИУ.

В общем случае, согласно ГОСТ 13377-55, надежность любого объекта определяется, как свойство этого объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующих заданным режимам и условиям использования, технического обслуживания, ремонта, хранения и транспортирования.

Основные свойства надежности изучаются теорией надежности, описывающей математические модели, методы и алгоритмы, направленные на изучение проблем предсказания, оценки и оптимизации различных показателей надежности.

В более общем смысле, «теория надежности устанавливает закономерности возникновения отказов устройств и методы их прогнозирования; ищет способы повышения надежности

изделий при конструировании и последующем изготовлении, а также приемы поддержания надежности во время их хранения и эксплуатации; разрабатывает методы проверки надежности при приемке больших партий продукции». Теория надежности вводит в рассмотрение количественные показатели качества систем.

В ГОСТ 15467-79, ГОСТ 13377-55, ГОСТ 27.002-89 приведены основные термины теории надежности:

- безотказность — свойство объекта непрерывно сохранять работоспособное состояние в течение некоторого времени или периода наработки;
- долговечность — свойство объекта сохранять работоспособное состояние при установленной системе технического обслуживания и ремонта;
- ремонтпригодность — свойство объекта, заключающееся в приспособленности к поддержанию и восстановлению работоспособного состояния путем технического обслуживания и ремонта;
- сохраняемость — свойство объекта сохранять в заданных пределах значения параметров, характеризующих способность объекта выполнять требуемые функции, в течение периода хранения и (или) транспортирования.

Вышеперечисленные свойства надежности характеризуют определенные технические состояния объекта. Различают пять основных видов технического состояния объектов:

- 1) исправное состояние, при котором объект соответствует всем требованиям нормативно-технической и (или) конструкторской (проектной) документации;
- 2) неисправное состояние, при котором объект не соответствует хотя бы одному из требований нормативно-технической и (или) конструкторской (проектной) документации;
- 3) работоспособное состояние, при котором значения всех параметров, характеризующих способность объекта выполнять заданные функции, соответствуют требованиям нормативно-технической и (или) конструкторской (проектной) документации;
- 4) неработоспособное состояние, при котором значения хотя бы одного параметра, характеризующего способность объекта выполнять заданные функции, не соответствует требованиям нормативно-технической и (или) конструкторской (проектной) документации;
- 5) предельное состояние, при котором дальнейшая эксплуатация объекта недопустима или нецелесообразна, либо восстановление его работоспособного состояния невозможно или нецелесообразно.

Переход объекта (изделия) из одного вышестоящего технического состояния в нижестоящее состояние обычно происходит вследствие событий, таких как повреждения или отказы.

Согласно ГОСТ 27.002-89, отказ — это событие, заключающееся в нарушении работоспособного состояния объекта; повреждение — событие, заключающееся в нарушении исправного состояния объекта при сохранении работоспособного состояния.

В ГОСТ 15467-79 введено еще одно понятие, отражающее состояние объекта, — дефект. Дефектом называется каждое отдельное несоответствие объекта установленным нормам или требованиям. Дефект отражает состояние, отличное от отказа.

Совокупность фактических состояний исследуемого объекта и возникающих событий, способствующих переходу в новое состояние, охватывает так называемый жизненный цикл (ЖЦ) объекта, который протекает во времени и имеет определенные закономерности, изучаемые в теории надежности.

В соответствии с определением отказа как события, заключающегося в нарушении работоспособности, предполагается, что до появления отказа объект был работоспособен. Отказ

может быть следствием развития не устраненных повреждений или наличия дефектов: царапин; потертости изоляции; небольших деформаций.

Оценка надежности представляет собой измерение количественных метрик атрибутов субхарактеристик в использовании, завершенности, устойчивости к дефектам, восстанавливаемости и доступности или готовности объекта.

Надежность технических систем определяется двумя главенствующими факторами: надежностью компонент и дефектами в конструкции, допущенными при проектировании или изготовлении. Эти же факторы учитываются и при оценке надежности АСОИУ, при этом определяющими являются ошибки, возникающие на стадии проектирования систем, поскольку надежность систем при их функционировании (физическая целостность) зависит от человеческого фактора и надежности носителя информации (жесткого магнитного диска, CD и т.д.). При этом надежность системы при обработке данных зависит от того, насколько безошибочно спроектирована и технически качественно реализована система. Следует учитывать и возможность самой системы влиять на бесперебойную работу вычислительной техники.

6.1.2. Особенности понятия надежность ПО АСОИУ

При использовании понятия надежности в области разработки ПО АСОИУ следует учитывать особенности и отличия объектов таких систем от традиционных технических систем:

- не для всех видов АСОИУ применимы понятия и методы теории надежности. Данные понятия и методы можно использовать только для АИС, функционирующих в реальном времени и непосредственно взаимодействующих с внешней средой;
- для повышения надежности комплексов программ особое значение имеют методы автоматического сокращения длительности восстановления и преобразования отказов в кратковременные сбои путем введения в программные компоненты элементов временной, программной и информационной избыточности;
- относительно редкое разрушение программных компонент и необходимость их физической замены приводит к принципиальному изменению понятий сбоя и отказа программ и к разделению их по длительности восстановления относительно некоторого допустимого времени простоя для функционирования АИС;
- непредсказуемость места, времени и вероятности проявления дефектов и ошибок, а также их редкое обнаружение при реальной эксплуатации достаточно надежных программных средств не позволяет эффективно использовать традиционные методы априорного расчета показателей надежности сложных систем, ориентированных на стабильные измеряемые значения надежности составляющих компонент;
- традиционные методы форсированных испытаний надежности систем путем физического воздействия на их компоненты не применимы для АСОИУ и их следует заменять методами форсированного воздействия информационных потоков внешней среды на систему.

Таким образом, с учетом вышеперечисленного можно сформулировать задачи теории и анализа надежности АСОИУ:

- формирование основных понятий, используемых при исследовании и применении показателей надежности программных средств;
- выявление и исследование основных факторов, определяющих характеристики надежности сложных программных комплексов;
- выбор и обоснование критериев надежности для комплексов программ различного типа и назначения;
- исследование дефектов и ошибок, динамики их изменения при отладке и сопровождении;

- исследование и разработка методов структурного построения АСОИУ, обеспечивающих необходимую надежность;
- исследование методов и средств контроля и защиты от искажений программ, вычислительного процесса и данных путем использования различных видов избыточности и защиты от помех;
- разработка методов и средств определения и прогнозирования характеристик надежности в жизненном цикле АСОИУ с учетом их функционального назначения, сложности, структурного построения и технологии разработки.

Решение этих задач позволяет обеспечить создание АСОИУ с заданными показателями надежности. Основными характеристиками надежности АСОИУ являются: отсутствие ошибок, устойчивость к ошибкам, возможность перезапуска системы.

Надежность любой программной системы тем выше, чем реже в ней происходят сбои, особенно такие, которые приводят к потере информации.

Надежная программа должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени. Быстрое реагирование системы на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньшее, чем порог между сбоем и отказом, характерные для корректно выполненных программ, обеспечивают ее высокую надежность. При этом и некорректная программа может функционировать абсолютно надежно.

В реальных условиях по разным причинам исходные данные могут попадать в области значений, вызывающих сбои, не проверенные при испытаниях, а также не заданные требованиями спецификации и технического задания. Если в этих ситуациях происходит достаточно быстрое восстановление и не фиксируется отказ, то такие события не влияют на основные показатели надежности — наработку на отказ и коэффициент готовности. Следовательно, надежность функционирования программ является понятием динамическим, проявляющимся во времени.

Непредсказуемость вида, места и времени проявления дефектов системы в процессе эксплуатации приводит к необходимости создания специальных, дополнительных систем оперативной защиты от непредумышленных, случайных искажений вычислительного процесса, программ и данных.

Системы оперативной защиты предназначены для выявления и блокирования распространения негативных последствий проявления дефектов и уменьшения их влияния на надежность функционирования ПО АСОИУ до устранения их первичных источников. Для этого в системе должна вводиться временная, программная и информационная избыточность, осуществляющая оперативное обнаружение дефектов функционирования, их идентификацию и автоматическое восстановление (рестарт) нормального функционирования АСОИУ. Надежность системы должна повышаться за счет средств обеспечения помехоустойчивости, оперативного контроля и восстановления функционирования программ и баз данных. Эффективность такой защиты зависит от используемых методов, координированности их применения и выделяемых вычислительных ресурсов на их реализацию. Надежность функционирования АСОИУ характеризуется устойчивостью, или способностью к безотказному функционированию, и восстанавливаемостью работоспособного состояния после произошедших сбоев или отказов.

Устойчивость системы зависит от уровня критичности неустраненных дефектов и ошибок и способности АСОИУ реагировать на их проявления так, чтобы это не отражалось на показателях надежности, что определяется эффективностью контроля данных, поступающих из внешней среды, и средств обнаружения аномалий функционирования АСОИУ.

Восстанавливаемость характеризуется полнотой и длительностью восстановления функционирования программ в процессе перезапуска системы (рестарта). Перезапуск должен обеспечивать возобновление нормального функционирования системы. На это требуются ресурсы ЭВМ и время. Поэтому полнота и длительность восстановления функционирования после сбоев отражают качество и надежность АСОИУ и возможность его использования по прямому назначению. Для определения надежности АСОИУ используется критерий длительности наработки на отказ, который определяется временем «работоспособного состояния системы между последовательными отказами или началами нормального функционирования системы после них».

Вероятностные характеристики этой величины также используются в качестве критериев надежности, учитывая возможность многократных отказов и восстановлений. Для оценки надежности восстанавливаемых систем важную роль играют «характеристики функционирования после отказа в процессе восстановления».

Основным показателем процесса восстановления является длительность восстановления и ее вероятностные характеристики. Этот показатель учитывает возможность многократных отказов и восстановлений. Обобщение характеристик отказов и восстановлений выражается в показателе коэффициента готовности, который отражает вероятность того, что система будет находиться в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы системы на достаточно большом интервале, содержащем отказы и восстановления.

Надежность ПО АСОИУ можно представить в виде следующей схемы (рис. 2).

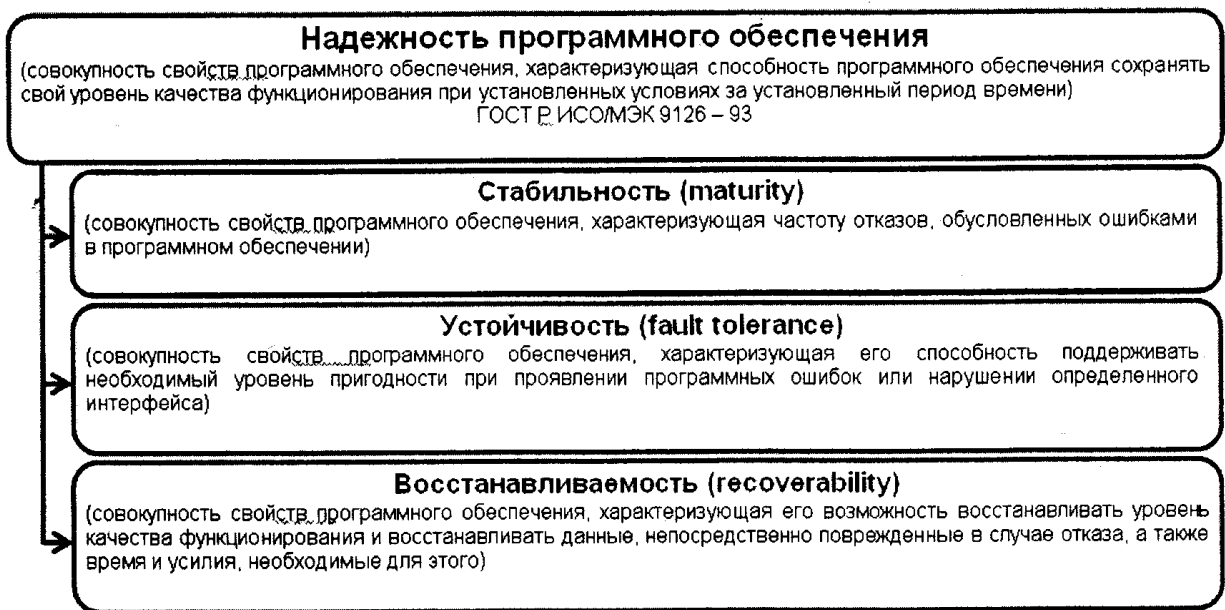


Рис. 2. Надежность программного обеспечения

Для оценки стабильности ПО можно использовать показатели для характеристики безотказности технических устройств (рис. 3).

6.1.3. Основные количественные показатели надежности ПО АСОИУ

1. *Вероятность безотказной работы $P(t_3)$* – это вероятность того, что в пределах заданной наработки, т.е. заданного времени работы системы t_3 , в часах, отказ системы не возникает.

2. *Вероятность отказа $Q(t_3)$* – вероятность того, что в пределах заданной наработки t_3 отказ системы возникает. Этот показатель обратный предыдущему, т.е.

$$Q(t_3) = 1 - P(t_3). \quad (2.1)$$

Их статистические оценки вычисляются по формулам:

$$Q(t_3) = \frac{n}{N_0}; \quad P(t_3) = 1 - \frac{n}{N_0},$$

где n - количество отказов за время испытаний;

N_0 - количество испытываемых объектов.

Пример. В течение 600 минут испытывались 10 автоматизированных рабочих мест (АРМ) системы. Зафиксировано 2 отказа.

Вероятность безотказной работы: $P(600) = 1 - 0,2 = 0,8$.

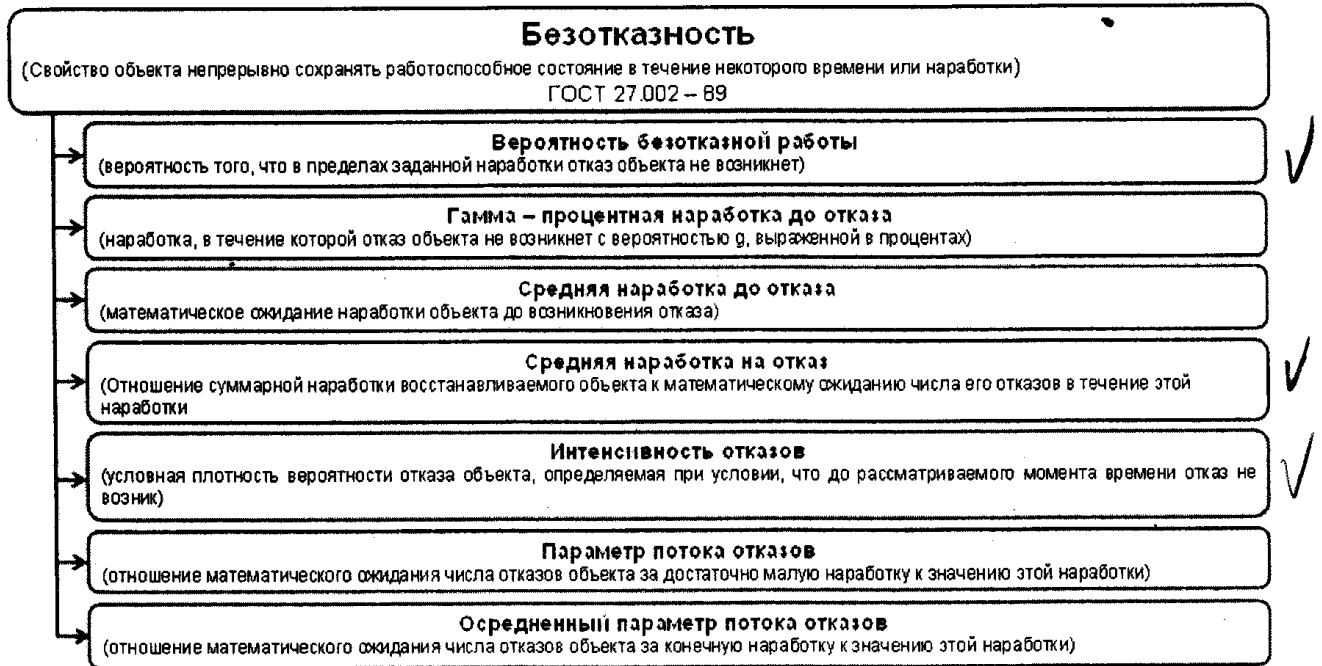


Рис. 3. Показатели безотказности

Вероятность отказа: $Q(600) = 2/10 = 0,2$.

3. *Интенсивность отказов системы* – это условная плотность вероятности возникновения отказов ПО в определенный момент времени при условии, что до этого времени отказ не возник.

$$\lambda(t) = \frac{f(t)}{P(t)}, \quad (2.2)$$

где $f(t)$ – плотность вероятности отказа в момент времени t .

$$f(t) = \frac{d}{dt} Q(t) = \frac{d}{dt} (1 - P(t)). \quad (2.3)$$

Существует следующая связь между интенсивностью отказов системы и вероятностью безотказной работы:

$$P(t) = \exp\left(-\int_0^t \lambda(t) dt\right) \quad (2.4)$$

В частном случае при

$$\begin{aligned} P(t) &= \exp(-\lambda t) \\ \lambda(t) &= \text{const.} \end{aligned} \quad (2.5)$$

Интенсивность отказов (λ) — один из наиболее важных в теоретическом и практическом отношении показателей, который определяет законы распределения вероятностей отказов. В результате многочисленных исследований и обработки статистических испытаний установлено, что

распределение вероятностей отказов подчиняется экспоненциальному закону надежности, параметром которого является величина λ . Таким образом, интенсивность отказов характеризует частоту появления отказов.

В практических целях пользуются статистической оценкой появления отказов [12], определяемой по формуле

$$\lambda = \frac{n}{N_0 \times \Delta t} \times \left[\frac{1}{\text{время}} \right],$$

где n – количество отказов, зафиксированных за время испытаний Δt ,

N_0 – количество испытываемых образцов.

Если выражение $\frac{n}{N_0}$ считать относительным количеством отказов за время Δt , то интенсивность отказов можно определить как относительное количество отказов, приходящееся на каждую единицу времени.

Пример. Два автоматизированных рабочих места (АРМ) информационной системы испытывались 100 часов. В течение этого времени было зафиксировано 20 отказов. Интенсивность отказов составит:

$$\lambda = \frac{20}{20 \times 100} = 0,1 \left[\frac{1}{\text{час}} \right]$$

4. *Средняя наработка на отказ T_i* – математическое ожидание времени работы ПО до очередного отказа:

$$T_i = \int_0^t t \cdot f(t) dt \quad (2.6)$$

Иначе среднюю наработку на отказ T_i можно представить следующим образом:

$$T_i = \frac{t_1 + t_2 + t_3 + \dots + t_n}{n} = \frac{1}{n} \cdot \sum_{i=1}^n t_i, \quad (2.7)$$

где t_i – время работы ПО между отказами, с.

n – количество отказов.

5. *Среднее время восстановления T_B* – математическое ожидание времени восстановления системы:

$$T_B = \frac{1}{n} \sum_{i=1}^n \tau_i$$

где n – число восстановлений, равное числу отказов;

τ_i – время, затраченное на восстановление (обнаружение, поиск причины и устранение отказа), в часах.

Показатель T_B можно определить на основании статистических данных, полученных для M однотипных восстанавливаемых объектов. Структура расчетной формулы та же:

$$T_B = \frac{\sum_{j=1}^M \tau_j}{\sum_{j=1}^M n_j},$$

где M – количество однотипных объектов, для каждого из которых определено общее время восстановления τ_j за заданное время наблюдений:

$$\tau_j = \sum_{i=1}^{n_j} \tau_{ij},$$

где τ_{ij} – время восстановления j -го объекта после i -го отказа; n_j – количество восстановления j -го объекта за время наблюдения, причем $1 \ll j \ll M$.

6. Коэффициент готовности K_r . Процесс функционирования объекта можно представить как последовательность чередующихся интервалов работоспособности и восстановления (простоя).

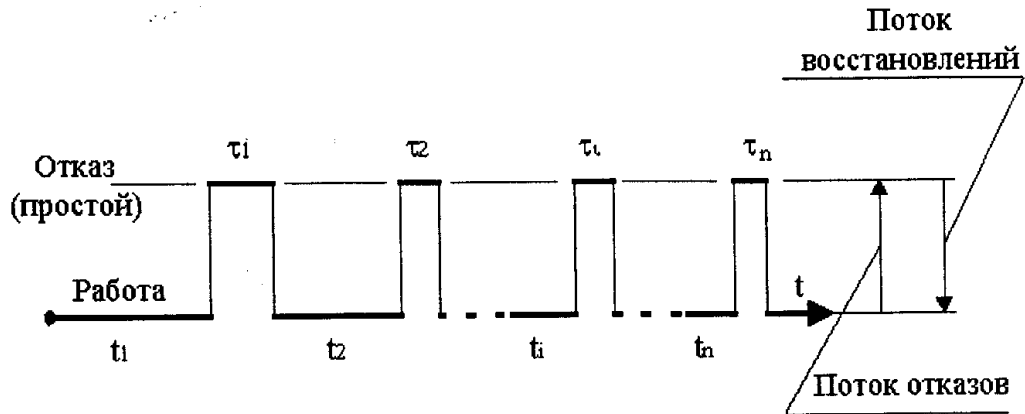


Рис. 2.4. График функционирования восстанавливаемого объекта:
 $t_1 \dots t_n$ – интервалы работоспособности; $\tau_1 \dots \tau_n$ – интервалы восстановления

Коэффициент готовности – это вероятность того, что объект окажется в работоспособном состоянии в произвольный момент времени, кроме планируемых периодов, в течение которых применение объекта по назначению не предусматривается. Этот показатель одновременно оценивает свойства работоспособности и ремонтпригодности объекта. Для вычисления этого показателя применяют следующую формулу:

$$K_r = \frac{\sum_{i=1}^n t_i}{\sum_{i=1}^n t_i + \sum_{i=1}^n \tau_i}.$$

7. Все приведенные показатели надежности ПО характеризуют наличие ошибок программы (производственных дефектов), но ни один из них не характеризует характер этих ошибок и возможные их последствия. Поэтому предлагается ввести новый показатель надежности ПО – *Средняя тяжесть ошибок (СТО)*. Вычисляется по формуле:

$$СТО = \frac{1}{Q} \cdot \sum_{i=1}^n b_i \cdot P_i \cdot z_i,$$

где Q – вероятность сбоя ПО;

b_i – функция принадлежности тяжести последствий ошибки, возникшей при i -м наборе входных данных к максимально тяжелым последствиям;

P_i – вероятность ввода i -го набора входных данных при эксплуатации ПО;

z_i – дихотомическая переменная, равная 1, если при i -ом наборе входных данных был зафиксирован сбой, и 0 в противном случае;

n – общее число наборов входных данных.

Значение показателя надежности СТО лежит на интервале $[0; 1]$. Чем ближе значение СТО к единице, тем тяжелее последствия ошибок ПО, и тем менее надежна программа. Близость СТО к нулю показывает незначительность последствий ошибок программы.

Введение нового показателя надежности ПО позволило различать по надежности программные продукты, вероятности сбоя которых имеют один и тот же порядок. К тому же, говоря о надежности ПО, пользователь желает получить не столько безошибочное ПО, сколько безопасное. А именно безопасность ПО характеризует СТО. Значение этого показателя субъективно и может быть различным для одного и того же программного продукта в зависимости от области его применения. Это объясняется тем, что при использовании конкретного ПО, например, для выполнения студенческих расчетов и для выполнения конструкторских расчетов в космической промышленности последствия ошибок программы - несопоставимы. В ряде случаев, если к ПО предъявляются жесткие требования, лучше оценивать максимальную тяжесть ошибок ПО.

Таким образом, оценивая вероятность сбоя ПО и СТО ПО, получаем многостороннюю оценку надежности ПО.

Вероятность безотказной работы $P(T_0)$ — это вероятность того, что объект сохранит работоспособность, т.е. не будет отказов в течение заданного интервала времени T_0 . При этом не имеет значения, сколько проработал объект к началу временного интервала. Вероятность безотказной работы и вероятность отказа образуют полную группу несовместных и противоположных событий, и соответственно справедливо следующее:

$$P(T_0) = 1 - Q(T_0); \quad Q(T_0) = 1 - P(T_0).$$

Тогда их статистические оценки будут вычислены следующим образом:

$$Q(T_0) = \frac{n}{N_0}; \quad P(T_0) = 1 - \frac{n}{N_0},$$

где n - количество отказов за время испытаний;

N_0 - количество испытываемых объектов.

Пример. В течение 600 минут испытывались 10 АРМов системы. Зафиксировано 2 отказа.

Вероятность безотказной работы: $P(600) = 1 - 0,2 = 0,8$.

Вероятность отказа: $Q(600) = 2/10 = 0,2$.

При оценке надежности программного обеспечения в настоящее время достаточно широкое распространение нашла статистическая модель Миллса, которая основана на известном из теории вероятностей гипергеометрическом законе распределения и использовании метода максимального правдоподобия для нахождения наиболее вероятного значения неизвестной величины [3].

Статистическая модель оценки надежности программного обеспечения Миллса основана на следующей *технологии* [3]:

- программа «засоряется» некоторым количеством известных, искусственных ошибок. Эти ошибки вносятся в программу таким образом, чтобы смоделировать случайную природу ошибок, оставшихся в программе, и чтобы вероятность обнаружения внесенных (искусственных) и собственных ошибок (все ещё находящихся в программе) при последующем тестировании была практически одинакова и зависела только от их количества;
- проводится тестирование программы, в результате которого обнаруживаются как внесенные, так и собственные ошибки.

Модели надежности программного обеспечения

Все модели надежности можно классифицировать по тому, какой из перечисленных процессов они поддерживают (предсказывающие, прогнозные, измеряющие и т.д.). Нужно отметить, что модели надёжности, которые в качестве исходной информации используют данные об интервалах между отказами, можно отнести и к измеряющим, и к оценивающим в равной степени. Некоторые модели, основанные на информации, полученной в ходе тестирования ПО, дают возможность делать прогнозы поведения ПО в процессе эксплуатации.

Аналитические модели дают возможность рассчитывать количественные показатели надежности, основываясь на данных о поведении программы в процессе тестирования (измеряющие и оценивающие модели).

Эмпирические модели базируются на анализе структурных особенностей программ. Они рассматривают зависимость показателей надёжности от числа межмодульных связей, количества циклов в модулях и т.д.

Часто эмпирические модели не дают конечных результатов показателей надёжности, однако они включены в классификационную схему, так как развитие этих моделей позволяет выявлять взаимосвязь между сложностью АСОД и его надежностью. Эти модели можно использовать на этапе проектирования ПО, когда осуществляется разбивка на модули и известна его структура.

Аналитические модели представлены двумя группами: динамические модели и статические. В динамических поведении ПО (появление отказов) рассматривается во времени. В статических моделях появление отказов не связывают со временем, а учитывают только зависимость количества ошибок от числа тестовых прогонов (по области ошибок) или зависимость количества ошибок от характеристики входных данных (по области данных).

Для использования динамических моделей необходимо иметь данные о появлении отказов во времени. Если фиксируются интервалы каждого отказа, то получается непрерывная картина появления отказов во времени (группа динамических моделей с непрерывным временем). С другой стороны, может фиксироваться только число отказов за произвольный интервал времени.

8. Модели надежности ПО. Динамическая модель Шумана.

Исходные данные для модели Шумана, которая относится к динамическим моделям дискретного времени, собираются в процессе тестирования АСОД в течение фиксированных или случайных временных интервалов. Каждый интервал - это стадия, на котором выполняется последовательность тестов и фиксируется некоторое число ошибок.

Модель Шумана может быть использована при определенном образе организованной процедуре тестирования. Использование модели Шумана предполагает, что тестирование поводится в несколько этапов. Каждый этап представляет собой выполнение на полном комплексе разработанных тестовых данных. Выявление ошибки регистрируется, но не исправляются. По завершении этапа на основе собранных данных о поведении ПО на очередном этапе тестирования может быть

использована модель Шумана для расчета количественных показателей надежности. При использовании модели Шумана предполагается, что исходное количество ошибок в программе постоянно, и в процессе тестирования может уменьшаться по мере того, как ошибки выявляются и исправляются.

Предполагается, что до начала тестирования в ПО имеется M ошибок. В течении времени тестирования τ в системе обнаруживается $\varepsilon_1(\tau)$ ошибок в расчете на одну команду в машинном языке.

Таким образом, удельное число ошибок на одну машинную команду, оставшуюся в системе после времени тестирования τ , равно:

$$\varepsilon_2(\tau) = \frac{M}{I} - \varepsilon_1(\tau) \quad (1)$$

где I - общее число машинных команд, которое предполагается постоянным в рамках этапа тестирования.

Если предположить, что значение функции частоты отказов $Z(t)$ пропорционально числу ошибок, оставшихся в программе после израсходованного на тестирование времени τ . Тогда можно записать:

$$Z(t) = C \cdot \varepsilon_2(\tau),$$

где C - некоторая константа,

t - время работы программного продукта (ПП) без отказов.

Тогда, если время работы программы без отказа t отсчитывается от точки $t = 0$, а τ остается фиксированным, то функция надежности, или вероятность безотказной работы на интервале времени от 0 до t , равна:

$$P(t, \tau) = \exp\left(-C \left(\frac{M}{I} - \varepsilon_1(\tau)\right) \cdot t\right) \quad (2)$$

$$t_{cp} = \frac{1}{C \cdot \left(\frac{M}{I} - \varepsilon_1(\tau)\right)} \quad (3)$$

Из величин, входящих в формулы (2) и (3), не известны начальное значение ошибок в программе M и коэффициент пропорциональности C . Для их определения прибегают к следующим рассуждениям.

Эти неизвестные можно оценить путем пропуска функционального теста в двух точках переменной оси отладки τ_a и τ_b , выбранных так, что $\varepsilon_1(\tau_a) < \varepsilon_1(\tau_b)$.

В процессе тестирования собирается информация о времени и количестве ошибок на каждом прогоне, т.е. общее время тестирования τ складывается из времени каждого прогона:

$$\tau = \tau_1 + \tau_2 + \tau_3 + \dots + \tau_n.$$

Предполагая, что интенсивность появления ошибок постоянна и равна λ , можно вычислить ее как число ошибок в единицу времени,

$$\lambda = \frac{\sum_{i=1}^n A_i}{\tau} \quad (4)$$

где A_i - количество ошибок на i - ом прогоне.

Тогда среднее время отказа t_{cp} , т.е. математическое ожидание экспоненциального закона распределения, можно определить по формуле

$$t_{cp} = \frac{\tau}{\sum_{i=1}^n A_i}. \quad (5)$$

Имея данные для двух различных моментов тестирования τ_a и τ_b , можно сопоставить уравнения (3) при τ_a и τ_b , т.е. получить следующие два выражения:

$$\frac{1}{\lambda_a} = \frac{1}{C \cdot \left(\frac{M}{I} - \varepsilon_1(\tau_a) \right)} \quad (6)$$

$$\frac{1}{\lambda_b} = \frac{1}{C \cdot \left(\frac{M}{I} - \varepsilon_1(\tau_b) \right)} \quad (7)$$

Из соотношений (6) и (7) можно найти неизвестные параметры C и M :

$$M^* = I \cdot \frac{\frac{\lambda_b}{\lambda_a} \cdot \varepsilon_1(\tau_a) - \varepsilon_1(\tau_b)}{\frac{\lambda_b}{\lambda_a} - 1} \quad (8)$$

$$C^* = \frac{\lambda_{\tau_a}}{\frac{M^*}{I} - \varepsilon_1(\tau_a)} \quad (9)$$

Получив неизвестные M^* и C^* , можно рассчитать надежность программы по формуле (2).

Рассмотрим несколько примеров.

Пример 1.

Программа содержит 2 000 командных строк, из них, до начала эксплуатации (после периода отладки), 15 командных строк содержат ошибки. После 20 дней работы обнаружена 1 ошибка. Найти среднее время безошибочной работы программы и интенсивность отказов программы при коэффициенте пропорциональности, равном 0,7.

Решение:

Данные для получения решения представим в следующем виде:

$I=$	2000
$M=$	15
$t=$	20
$x=$	1

$$C=0,7$$

Для получения решения воспользуемся формулами (1), (2) и (3).

В результате получим:

$$\varepsilon_1(20) = \frac{1}{2000} = 0,0005;$$

$$\varepsilon_2(20) = \frac{15}{2000} - 0,0005 = 0,007;$$

$$P(0, 20) = \exp\left(-0,7\left(\frac{15}{2000} - 0,0005\right) \cdot 20\right) = \exp(-0,098) = 0,9066;$$

$$t_{cp} = \frac{1}{0,7\left(\frac{15}{2000} - 0,0005\right)} = \frac{1}{0,7 \cdot 0,007} = 204,0816;$$

$$\lambda = \frac{1}{204,0816} = 0,0049 \text{ - интенсивность отказов.}$$

Пример 2.

На условиях примера 1 определить вероятность безошибочной работы программы в течение 90 суток.

$$I=2000$$

$$M=15$$

$$t=90$$

$$x=1$$

$$C=0,7$$

Выполнив формулу (2), получим следующий результат:

$$P(0, 90) = \exp\left(-0,7\left(\frac{15}{2000} - 0,0005\right) \cdot 90\right) = \exp(-0,441) = 0,643393.$$

Пример 3.

Определить первоначальное количество возможных ошибок в программе, содержащей 2 000 командных строк, если в течение первых 60 суток эксплуатации было обнаружено 2 ошибки, а за последующие 40 суток была обнаружена одна ошибка. Определить T_0 – среднее время безошибочной работы, соответствующее первому и второму периоду эксплуатации программы и коэффициент пропорциональности.

I=	2000
t1=	60 суток
t2=	100 суток
x1=	2 ош.
x2=	3 ош.
T0=	30 33,33333

Интенсивности

отказов:

$$\lambda_1=0,033333$$

$$\lambda_2=0,03$$

$$C=6,666667$$

$$E_1(t_1)=0,001$$

$$E_2(t_2)=0,0015$$

$$M=12$$

$L2/L1=$ 0,9

9. Модели надежности ПО. Статическая модель Миллса.

Статические модели принципиально отличаются от динамических прежде всего тем, что в них не учитывается время появления ошибок в процессе тестирования и не используется никаких предположений о поведении функции риска. Эти модели строятся на твердом статическом фундаменте.

Использование этой модели предполагает необходимость перед началом тестирования искусственно вносить в программу (засорять) некоторое количество известных ошибок. Ошибки вносятся случайным образом и фиксируются в протоколе искусственных ошибок. Специалист, проводящий тестирование, не знает ни количества ошибок, ни характера внесенных ошибок до момента оценки показателей надежности по модели Миллса. Предполагается, что все ошибки (как естественные, так и искусственно внесенные) имеют равную вероятность быть найденными в процессе тестирования.

Тестируя программу в течение некоторого времени, собирается статистика об ошибках. В момент оценки надежности по протоколу искусственных ошибок все ошибки делятся на собственные и искусственные.

Пусть в процессе тестирования обнаружено n исходных ошибок и v из S рассеянных ошибок. Тогда оценка N - первоначальное число ошибок в программе - составит

$$N = \frac{S \cdot n}{v}. \quad (1)$$

Вторая часть модели связана с проверкой гипотезы выражения и тестирования N .

Предположим, что программа содержит K собственных ошибок и S внесенных (рассеянных) ошибок.

Будем тестировать программу до тех пор, пока не обнаружим все рассеянные ошибки. В то же время количество обнаруженных исходных ошибок накапливается и запоминается. Пусть их количество равно n .

Тогда по формуле Миллса считаем, что первоначально в программе было $N=n$.

Вероятность, с которой можно высказать такое предположение, можно рассчитать по следующему соотношению:

$$C = \begin{cases} 1, & \text{если } n > K \\ \frac{S}{S + k + 1}, & \text{если } n \leq K \end{cases} \quad (2)$$

Таким образом, это есть вероятность того, что в программе содержится K собственных ошибок.

Величина C является мерой доверия к модели и показывает вероятность того, насколько правильно найдено значение N . Эти два связанных между собой по смыслу соотношения образуют полезную модель ошибок: первое предсказывает возможное число первоначально имевшихся в программе ошибок, а второе используется для установления доверительного уровня прогноза.

Формула для расчета C в случае, когда обнаружены не все искусственно рассеянные ошибки, модифицирована таким образом, что оценка может быть выполнена после обнаружения v ($v \leq S$) рассеянных ошибок:

$$C = \begin{cases} 1, & \text{если } n > K \\ \frac{C_S^{v-1}}{C_{S+K+1}^{k+v}}, & \text{если } n \leq K \end{cases}, \quad (3)$$

где числитель и знаменатель формулы (3) при $n \leq K$ являются биномиальными коэффициентами.

Например, если утверждается, что в программе нет ошибок, а к моменту оценки надежности обнаружено 5 из 10 рассеянных ошибок и не обнаружено ни одной собственной ошибки, то вероятность того, что в программе действительно нет ошибок, будет равна:

$$C = \frac{C_{10}^4}{C_{10+1}^{1+4}} = \frac{10! \cdot 5! \cdot 6!}{4! \cdot 6! \cdot 11!} = \frac{5}{11} \approx 0,45.$$

Если при тех же исходных условиях оценка надежности производится в момент, когда обнаружены 8 из 10 искусственных ошибок, то вероятность того, что в программе не было ошибок, увеличивается до 0.73.

В действительности модель Миллса можно использовать для оценки N после каждой найденной ошибки. Предлагается во время всего периода тестирования отмечать на графике число найденных ошибок и текущее значение для N .

Достоинством модели является простота применения математического аппарата, наглядность и возможность использования в процессе тестирования.

Однако она не лишена и ряда недостатков, самые существенные из которых - это необходимость внесения искусственных ошибок (этот процесс плохо формализуется) и достаточно вольное допущения величины K , которое основывается исключительно на интуиции и опыте человека, проводящего оценку, т.е. допускается большое влияние субъективного фактора.

Пример 4.

Предположим, что в программе имеется 3 собственных ошибки. Внесём ещё 6 ошибок случайным образом.

В процессе тестирования было найдено:

- 1) 6 ошибок из рассеянных и 2 собственных;
- 2) 5 ошибок из рассеянных и 2 собственных;
- 3) 5 ошибок из рассеянных и 4 собственных.

Найти надёжность по модели Миллса - C .

Решение.

- 1) $K=3$; $S=6$; $n=2$; $v=6$.

Так как $n=2 \leq K=3$ и $S=6 \neq v=6$, то применяем нижнюю строку формулы (2):

$$C = \frac{S}{S + K + 1} = \frac{6}{6 + 3 + 1} = 0,6.$$

- 2) $K=3$; $S=6$; $n=2$; $v=5$.

Так как $n=2 \leq K=3$ и $S=6 \neq v=5$, то применяем нижнюю строку формулы (3):

$$C = \frac{C_S^{v-1}}{C_{(S+K+1)}^{(K+v)}} = \frac{C_6^{(5-1)}}{C_{(6+3+1)}^{(3+5)}} = \frac{C_6^4}{C_{10}^8} = \frac{6! \cdot 8! \cdot 2!}{4! \cdot 2! \cdot 10!} = \frac{1}{3} \approx 0,333.$$

- 3) $K=3$; $S=6$; $n=4$; $v=5$.

Так как $n=4 > K=3$ и $S=6 > v=5$, то применяем верхнюю строку формулы (3):

$$C=1$$

10. Модели надежности ПО. Простая интуитивная модель, модель Коркорена.

Простая интуитивная модель

Использование этой модели предполагает проведение тестирования двумя группами программистов (или двумя программистами в зависимости от величины программы) независимо друг от друга, использующими независимые тестовые наборы. В процессе тестирования каждая из групп фиксируют все найденные ею ошибки.

Пусть первая группа обнаружила n_1 ошибок, вторая n_2 , n_{12} - это число ошибок, обнаруженных как первой, так и второй группой.

Обозначим через N неизвестное количество ошибок, присутствующих в программе до начала тестирования. Тогда можно эффективность тестирования каждой из групп определить как

$$E_1 = \frac{n_1}{N}, \quad E_2 = \frac{n_2}{N}$$

Эффективность тестирования можно интерпретировать как вероятность того, что ошибка будет обнаружена. Таким образом, можно считать, что первая группа обнаруживает ошибку в

программе с вероятностью $P_1 = \frac{n_1}{N}$, вторая - с вероятностью $P_2 = \frac{n_2}{N}$. Тогда вероятность p_{12} того,

что ошибка будет обнаружена обеими группами, можно принять равной $\frac{n_{12}}{N}$. С другой стороны, так

как группы действуют независимо друг от друга, то $p_{12} = p_1 p_2$. Получаем: $\frac{n_{12}}{N} = \frac{n_1}{N} \cdot \frac{n_2}{N}$

Отсюда получаем оценку первоначального числа ошибок программы:

$$N = \frac{n_1 n_2}{n_{12}}$$

Пример 5.

В процессе тестирования программы 1-я группа нашла 15 ошибок, 2-я группа нашла 25 ошибок, общих ошибок было 5. Определить надёжность по простой интуитивной модели.

$n_1 =$	15	ош
$n_2 =$	25	ош
общ.ош=	5	ош

N=	75	
p1=	0,2	
p2=	0,333333	
p12=	0,066667	

Модель Коркорэна

Применение модели предполагает знание следующих ее показателей:

- модель содержит изменяющуюся вероятность отказов для различных источников ошибок и соответственно разную вероятность их исправления;
- в модели используются такие параметры, как результат только N испытаний, в которых наблюдается N_i ошибок i-го типа;
- ☐ выявление в ходе N испытаний ошибки i-го типа появляется с вероятностью a_i .

Показатель уровня надежности R вычисляют по следующей формуле:

$$R = \frac{N_0}{N} + \sum_{i=1}^k \frac{Y_i * (N_i - 1)}{N},$$

где N_0 - число безотказных (или безуспешных) испытаний, выполненных в серии из N испытаний,
k - известное число типов ошибок, a_i — вероятность выявления при тестировании ошибки i-го типа,
 Y_i - вероятность появления ошибок, при $N_i > 0$, $Y_i = a_i$, при $N_i = 0$, $Y_i = 0$.

Пример 6.

Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки	Вероятность появления ошибки a_i	Число появления ошибок N_i при испытании	Y_i	$(Y_i * (N_i - 1)) / N$
------------	---------------------------------------	--	-------	-------------------------

1. Ошибки вычисления	0,09	5	0,09	0,0036
2. Логические ошибки	0,26	25	0,26	0,0624
3. Ошибки ввода/вывода	0,16	3	0,16	0,0032
4. Ошибки манипулирования данными	0,18	0	0	0
5. Ошибки сопряжения	0,17	11	0,17	0,017
6. Ошибки определения данных	0,08	3	0,08	0,0016
7. Ошибки в БД	0,06	4	0,06	0,0018

Оценить надёжность по модели Коркорэна.

Исходные данные:	
N=	100
N0=	20
R=	0,2896

Пример 7. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

Тип ошибки, i	Вероятность появления ошибки. a_i	Число появления ошибок N_i при испытании	Y_i	$(Y_i * (N_i - 1)) / N$
1	0,09	8	0,09	0,0063
2	0,26	0	0	0
3	0,17	4	0,17	0,0051
4	0,2	25	0,2	0,048
5	0,8	25	0,8	0,192
6	0,08	3	0,08	0,0016
7	0,16	5	0,16	0,0064

Оценить надёжность программы по модели Коркорэна.

Исходные данные:	
N=	100
N0=	20
R=	0,4594

11. Постановка и модель решения прямой задачи определения надёжности программного комплекса АСОИУ.

Сложность вновь создаваемых программных комплексов, высокая стоимость и сравнительно низкий уровень качества их разработки не позволяют исключить возможность возникновения ошибок, приводящих к нарушению работоспособности вычислительных средств и снижению их производительности. Поэтому возникает необходимость в получении объективных оценок надёжности создаваемого программного комплекса.

Используя термин «надёжность программного комплекса», аналогично термину «надёжность аппаратуры», следует помнить, что отказы программного комплекса, как результат проявления ошибок, имеют совершенно другую физическую природу, чем отказы техники. Однако это не является причиной невозможности использования некоторых терминов и показателей надёжности техники при исследовании надёжности программного комплекса. В частности, это оправдывается и необходимостью решения задачи распределения ресурсов или затрат между самими вычислительными средствами и их программным обеспечением при достижении заданного значения показателя надёжности.

Одним из важнейших общих показателей надёжности, представляющих интерес для практики, является вероятность безошибочного функционирования программного обеспечения.

Для анализа показателей надёжности сложного программного комплекса представим его в виде совокупности менее сложных составляющих, которыми могут быть программные модули. Программный модуль, в свою очередь, может быть разделен на более мелкие части – процедуры и т.д. Таким образом, программный модуль можно рассматривать как аналог элемента расчета теории надёжности. Обычно он представляет собой логически самостоятельную программу и вводится, исходя из представлений исследователя.

При исследовании надёжности функционирования программного комплекса можно решать две задачи.

Первая задача – по заданной структуре программного комплекса, состоящего из некоторой известной совокупности программных модулей, имеющих известные показатели надёжности, надо найти показатель надёжности программного комплекса. Эту задачу назовем прямой задачей.

Вторая задача – достижение экстремального (максимального или минимального) значения показателя надёжности при ограничениях на ресурсы, в качестве которых выступают время, стоимость и др. Или может решаться задача минимизации величины ограничения при достижении

требуемого (заданного) значения показателя надежности. Любую из этих задач назовем обратной задачей.

6.2.3.1. Постановка и модель решения прямой задачи

Рассмотрим вначале первую задачу. Итак, дан программный комплекс, состоящий из M отдельных модулей, соединенных между собой. По структуре программного комплекса можно построить стохастический граф, содержащий $M + 2$ вершин. Вершина 0 означает исток – начало программного комплекса, а вершина $M + 1$ – сток графа, т.е. завершение работы этого комплекса. Каждый программный модуль вызывается на решение с заданной вероятностью, исходя из цели функционирования программного комплекса или значений исходных данных. Задача заключается в нахождении вероятности безошибочных решений задачи программного комплекса, если известны вероятности безошибочных решений задач всех программных модулей.

Рассмотрим матрицу $G = G(t)$, $t = (t_0, t_1, \dots, t_{M+1})$.

Элементами этой матрицы являются произведения $p_{ij} \cdot P_i(t_i)$, $i, j = (0, 1, \dots, M + 1)$, где p_{ij} – вероятность перехода от программного модуля с номером i к программному модулю с номером j , а $P_i(t_i)$ – вероятность безошибочного функционирования программного модуля с номером i в течение времени t_i .

Согласно [1 и 2] вероятность безошибочной работы модуля программного обеспечения можно вычислить по следующей формуле:

$$P(t_i, \tau_i) = e^{-\lambda_i \cdot t_i} \cdot e^{-\nu_i \tau_i}, \quad (1)$$

где λ_i – интенсивность проявления ошибки модуля с номером i , а ν_i – интенсивность отладки модуля с номером i ; t_i и τ_i – время вычислений и отладки модуля с номером i . Обычно $\lambda_i = 1/T_i$, T_i – начальное среднее время безошибочной работы модуля с номером i ; $\nu_i = K_i / (N_{ошиб}^i \cdot T_i)$, K_i – коэффициент сжатия времени отладки (тестирования) по сравнению со временем вычислений, $N_{ошиб}^i$ – первоначальное предполагаемое число ошибок в модуле с номером i .

([1] Mysa J. A theory of software reliability and its application.// IEEE Trans. OnsoftwareEng., vol. SE-1, sept. 1975. – P.312-327.

[2] Кузнецов В.В, Смагин В.А. Прямая и обратная задачи надёжности сложных программных комплексов.// Надёжность и контроль качества. – 1997. – № 10. – С. 56-62.)

Так как вершины стохастического графа с номерами 0 и $M + 1$ – это фиктивные вершины, то полагаем, что время нахождения в них равно нулю, а вероятность безошибочной работы равна единице.

Введем понятие шага, подразумевая под ним единичный переход от одного программного модуля к другому. Чтобы найти вероятности безошибочной работы за два шага, нужно просуммировать с соответствующими вероятностями произведения вероятностей по всем путям, содержащим две вершины (одна из них нулевая). Это достигается возведением матрицы G в квадрат. При возведении матрицы G в куб получаем вероятности безошибочного функционирования за три шага и т.д.

Построим следующую матрицу T :

$$T = I + G(t) + G^2(t) + \dots = I(I - G(t))^{-1}, \quad (2)$$

где I – единичная матрица.

Элемент матрицы T с номером $(0, M + 1)$ представляет собой выражение для вероятности безошибочной работы всего программного комплекса с учетом всех возможных последовательностей вызовов отдельных программных модулей.

В соответствии с правилами вычисления значений элементов обратной матрицы, выражение для вероятности безошибочной работы программного комплекса можно представить в следующем виде:

$$Y(t) = \frac{Q(t)}{R(t)}, \quad (3)$$

где $Q(t)$ - алгебраическое дополнение элемента с номером $(M + 1, 0)$ матрицы $(I - G(t))$, а $R(t)$ - главный определитель матрицы $(I - G(t))$.

Выполнив указанные преобразования, получим искомое выражение для вероятности безошибочного функционирования программного комплекса с учетом задействования всех возможных маршрутов вычислений.

Рассмотрим следующий пример 1.

Пусть задан следующий стохастический граф, отображающий работу программного комплекса из трех программных модулей (рис.1).

Для данного программного комплекса известны следующие его параметры:

- коэффициенты сжатия времени отладки (тестирования) по сравнению со временем вычислений $K_1 = K_2 = K_3 = 1$;

- время вычислений программными модулями $t_1 = 1c$; $t_2 = 7c$; $t_3 = 10c$;

- первоначальное (предполагаемое) число ошибок в модулях $N_{ошиб}^1 = 10$; $N_{ошиб}^2 = 5$; $N_{ошиб}^3 = 3$;

- интенсивности проявления ошибки в модулях $\lambda_1 = 0,01$ 1/с; $\lambda_2 = 0,02$ 1/с;

$\lambda_3 = 0,03$ 1/с;

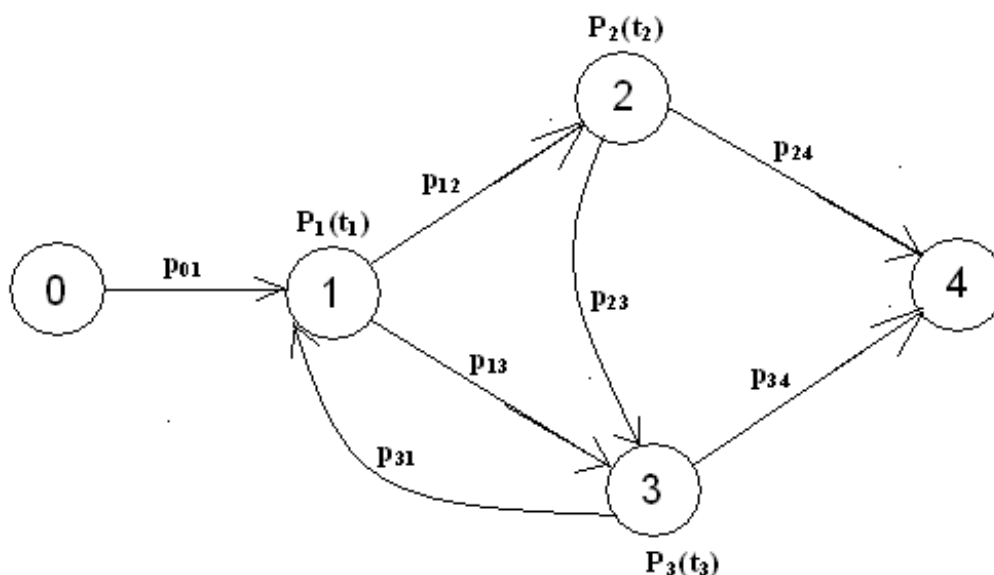


Рис. 1 Стохастический граф программного комплекса

- вероятности перехода от одного модуля к другому $p_{01} = 1$; $p_{12} = 0,7$; $p_{13} = 0,3$;

$p_{23} = 0,6$; $p_{24} = 0,4$; $p_{31} = 0,8$; $p_{34} = 0,2$;

- вероятности безошибочной работы программных модулей вычислим по формуле (1), полагая, что время отладки модуля с номером $i = 1, 2, 3$ для простоты вычислений равно нулю:

$$P_1(t_1 = 1, \tau_1 = 0) = e^{-0,011} = e^{-0,01} = 0,9901;$$

$$P_2(t_2 = 7, \tau_2 = 0) = e^{-0,027} = e^{-0,14} = 0,8694;$$

$$P_3(t_3 = 10, \tau_2 = 0) = e^{-0,0310} = e^{-0,3} = 0,7408.$$

Матрица $G(t)$ размерности 5×5 для данного примера имеет следующий вид:

$$G(t) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & p_{12}P_1 & p_{13}P_1 & 0 \\ 0 & 0 & 0 & p_{23}P_2 & p_{24}P_2 \\ 0 & p_{31}P_3 & 0 & 0 & p_{34}P_3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0,69307 & 0,29703 & 0 \\ 0 & 0 & 0 & 0,52164 & 0,34776 \\ 0 & 0,59264 & 0 & 0 & 0,14816 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Построим вначале матрицу $T^{-1} = (I - G(t))$:

$$T^{-1} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -p_{12}P_1 & -p_{13}P_1 & 0 \\ 0 & 0 & 1 & -p_{23}P_2 & -p_{24}P_2 \\ 0 & -p_{31}P_3 & 0 & 1 & -p_{34}P_3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -0,69307 & -0,29703 & 0 \\ 0 & 0 & 1 & -0,52164 & -0,34776 \\ 0 & -0,59264 & 0 & 1 & -0,14816 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Элемент матрицы $T = (I - G(t))^{-1}$ с номером (0, 4) согласно формуле (3) равен отношению алгебраического дополнения элемента (4, 0) матрицы $(I - G(t))$ и главного определителя этой матрицы. Раскрывая эти определители, получим следующее выражение:

$$\begin{aligned} Y(t) = P(t, 0) &= \frac{p_{12}P_1p_{23}P_2p_{34}P_3 + p_{12}P_1p_{24}P_2 + p_{13}P_1p_{34}P_3}{1 - p_{13}P_1p_{31}P_3 - p_{12}P_1p_{23}P_2p_{31}P_3} = \\ &= \frac{0,7 \cdot 0,9901 \cdot 0,6 \cdot 0,8694 \cdot 0,2 \cdot 0,7408 + 0,9901 \cdot 0,4 \cdot 0,8694 + 0,3 \cdot 0,9901 \cdot 0,2 \cdot 0,7408}{1 - 0,3 \cdot 0,9901 \cdot 0,8 \cdot 0,7408 - 0,7 \cdot 0,9901 \cdot 0,6 \cdot 0,8694 \cdot 0,8 \cdot 0,7408} \approx 0,56. \end{aligned} \quad (4)$$

Таким образом, для данного примера вероятность безошибочного решения задачи рассматриваемым программным комплексом равна 0,56, т.е. чуть больше половины.

Рассмотрим теперь вторую, обратную задачу. Найдем минимальное время отладки программного

комплекса $\tau = \sum_{i=0}^{M+1} \tau_i$, при котором вероятность безошибочного решения задачи $P(t, \tau) \geq P_{зад}$. При

этом будем предполагать, что вероятность безошибочного решения задачи можно определить по формуле (1). Для нахождения решения воспользуемся методом неопределенных множителей Лагранжа.

Функция Лагранжа для данного примера имеет следующий вид:

$$F(t, \tau, \gamma) = \sum_{i=0}^{M+1} \tau_i + \gamma(P(t, \tau) - P_{зад}), \quad (5)$$

где γ - множитель Лагранжа.

Дифференцируя выражение (5) по аргументам τ_i и γ , получим следующую систему уравнений:

$$\begin{cases} \frac{\partial F(t, \tau, \gamma)}{\partial \tau_i} = 0, \quad i = 0, 1, 2, \dots, M+1; \\ P(t, \tau_1, \tau_2, \dots, \tau_M, \gamma) - P_{\text{зад}} = 0. \end{cases} \quad (6)$$

Решив систему уравнений (6) относительно $\tau_i = \tau_i^0$, получим $\tau^0 = \sum_{i=0}^{M+1} \tau_i^0$.

Рассмотрим еще одну обратную задачу. Необходимо найти максимальное значение вероятности безошибочного функционирования программного

$$F(t, \tau, \gamma) = P(t, \gamma) + \gamma \left(\sum_{i=0}^{M+1} \tau_i - \tau_{\text{зад}} \right). \quad (7) \text{ комплекса при заданном}$$

времени его отладки. Для этого будем искать максимум функции $P(t, \tau)$ при заданном времени отладки $\tau_{\text{зад}}$. Функция Лагранжа в данном случае имеет следующий вид:

Дифференцируя выражение (7) по переменным $\tau_i, i = 0, 1, 2, \dots, M+1$ и γ и приравняв их нулю, получим следующую систему уравнений:

$$\begin{cases} \frac{\partial F(t, \tau, \gamma)}{\partial \tau_i} = 0, \quad i = 0, 1, 2, \dots, M+1; \\ \sum_{i=0}^{M+1} \tau_i - \tau_{\text{зад}} = 0. \end{cases} \quad (8)$$

Решив систему уравнений (8), можно найти искомые значения τ_i^0 , а подставляя их в выражение для $P(t, \tau)$, можно найти максимальное значение вероятности безошибочного функционирования программного комплекса.

Рассмотрим следующий пример 2. Пусть задано значение вероятности для программного комплекса $P_{\text{зад}} \geq 0,999$. Требуется найти времена отладок каждого модуля и суммарное минимальное время отладки программного комплекса.

Поступая, как описано в п. «Обратная задача», находим $\tau_1^0 = 3460$ с; $\tau_2^0 = 1871$ с; $\tau_3^0 = 848$ с. Отсюда следует, что время отладки всего программного комплекса равно $\tau = 3460 + 1871 + 848 = 6179$ с.

Рассмотрим следующий пример 3. Пусть задано время отладки программного комплекса $\tau_{\text{зад}} = 6000$ с. Требуется найти времена отладок каждого модуля $\tau_i^0, i = 1, 2, 3$ и максимальное значение вероятности его безошибочного функционирования при тех же данных задачи 1. В результате получим:

$$\tau_1 = 3327 \text{ с}; \tau_2 = 1838 \text{ с}; \tau_3 = 835 \text{ с}; \max P(t, \tau) = 0,999.$$

Приведенные числовые примеры иллюстрируют возможности использования рассмотренных методов в решении прикладных задач, связанных с анализом надежности программных комплексов и обеспечением требуемых значений показателей их надежности при отладке.

12. Классификация технической документации АСОИУ. Структура предпроектной документации.

Требования к содержанию документов, разрабатываемых при создании АС, установлены методическими указаниями по информационной технологии РД 50-34.698-90, а также государственными стандартами Единой системы программной документации (ЕСПД), Единой системы конструкторской документации (ЕСКД). Системы проектной документации для строительства (СПДС) и ГОСТ 34.602. Виды и комплектность документов регламентированы ГОСТ 34.201.

Техническая документация на АС в общем случае может быть классифицирована следующим образом.

1. Техническая документация – комплекс взаимосвязанных документов, в которых полностью описаны все решения по созданию и эксплуатации АС.

2. Предпроектная документация – часть технической документации, содержащая обоснование целесообразности создания АС, требования к ней и порядок выполнения работ по ее созданию.

3. Проектная документация – часть технической документации на АС, содержащая проектные решения по созданию и эксплуатации АС.

4. Приемосдаточная документация – часть технической документации на АС, подтверждающая соответствие АС требованиям технического задания и готовность АС к эксплуатации.

5. Эксплуатационная документация на АС – часть проектной документации на АС, предназначенная для обеспечения ее функционирования.

6. Проектно-сметная документация на АС – часть проектной документации на АС, входящая в состав проектов промышленного строительства.

7. Общесистемная документация проекта АС – часть проектной документации на АС, содержащая описание, обоснование и оценку принятых решений по АС в целом и подготовке объекта к вводу АС в эксплуатацию.

8. Документация функциональной части АС – часть проектной документации, содержащая решения по функциям управления объектом, функциональной структуре и постановкам задач.

9. Документация организационного обеспечения АС – часть проектной документации на АС, содержащая решения по организационной структуре и инструкции персоналу.

10. Документация информационного обеспечения АС – часть проектной документации на АС, содержащая решения по информационной базе, системе классификации и кодирования и технологическому процессу обработки информации в АС.

11. Документация технического обеспечения АС – часть проектной документации, содержащая решения по комплексу технических средств проектируемой АС.

12. Документация программного обеспечения АС – часть проектной документации, содержащая решения по применяемым программам и программному обеспечению в целом.

13. Документация математического обеспечения АС – часть проектной документации, содержащая описание применяемых математических моделей и алгоритмов.

9.2. Предпроектная документация

Согласно названию, *предпроектными* считаются документы, разрабатываемые до начала проектных действий по созданию АС. Эти документы составляются для выявления и обоснования необходимости разработки системы, формулирования требований к ней, а также для согласований условий создания и внедрения.

9.2.1. Материалы обследования объекта автоматизации

Сведения, собранные разработчиком в результате экспресс- или полного обследования объекта, представляются в документах двух видов:

Отчет - в нем приводятся первичные данные, характеризующие состояние информационной системы объекта «как есть», т. е. на момент обследования. Форма и структура отчета об обследовании не регламентированы, но многие разработчики при его написании руководствуются требованиями к отчетам по научно-исследовательским работам, изложенным в ГОСТ 7.32-2001 «Система стандартов

по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления». Рекомендации по содержанию отчета приведены в приложении 1 к РД 50-34.698-90.

Ранее эти вопросы были рассмотрены в разделе 2.2. курса лекций (см. этап 1.3 «Оформление отчета о выполненной работе и заявки на разработку АСОИУ»).

Технико-экономическое обоснование (ТЭО). Технико-экономическое обоснование представляет собой документально оформленные результаты маркетинговых и технико-экономических исследований, обосновывающих целесообразность и возможности реализации инвестиционного проекта, выбор наиболее эффективных организационных, технических и экономических решений для ввода в действие новых или реконструкции и модернизации действующих производственных мощностей. Документ имеет аналитический характер. В нем исследуется эффективность автоматизации конкретных функций обработки информации, рассчитываются необходимые затраты и оцениваются различные варианты АС. ТЭО позволяет разработчику определить привлекательность и прибыльность проекта АС, а заказчику - принять решение о целесообразности создания предлагаемых вариантов системы. Для составления технико-экономического обоснования могут применять различные (в том числе и фирменные) методики, поскольку универсальной и обязательной к применению методики в настоящее время не существует.

Материалы обследования объекта автоматизации в совокупности с профессиональным опытом разработчика и результатами анализа аналогов прототипов, используются для формулирования концепции создания автоматизированной системы, отражаемой в *техническом задании* на АС.

9.2.2. Техническое задание

Мы уже отмечали важнейшую роль технического задания как документа, определяющего технико-тактические требования к создаваемой АС. Содержание ТЗ отражает предложенную разработчиком концепцию создания автоматизированной системы, и в комплексе с договорной документацией определяет условия ее построения и внедрения.

Требования к составу и содержанию технического задания на АС определены в Государственном стандарте ГОСТ 34.602-89. Этот стандарт содержит указания по составлению каждого из девяти обязательных разделов ТЗ, а также правила его оформления и внесения изменений. Подробные рекомендации по написанию технического задания на создание АС были даны ранее в разделе 3.1 «Разработка и утверждение задания на создание АС».

9.2.3. Договорная документация

Согласно ст. 420 ГК РФ, договором признается соглашение двух или нескольких лиц об установлении, изменении или прекращении гражданских прав и обязанностей. Договор должен соответствовать обязательным для сторон правилам, установленным законом и иными правовыми актами (императивным нормам), действующим в момент его заключения. Если после заключения договора принят закон, устанавливающий обязательные для сторон правила иные, чем те, которые действовали при заключении договора, условия заключенного договора сохраняют силу, кроме случаев, когда в законе установлено, что его действие распространяется на отношения, возникшие из ранее заключенных договоров (ст. 422 ГК РФ).

Договор на создание АС или на выполнение отдельных работ жизненного цикла АС заключается, как правило, в письменной форме. В договоре определяется объект автоматизации и формулируется конкретный перечень подлежащих выполнению работ, а также устанавливаются

права и обязанности каждой стороны. По соглашению сторон в договоре могут перечисляться чрезвычайные обстоятельства (вызванные действием непреодолимой силы), наступление которых может повлечь досрочное прекращение исполнения договора без каких-либо юридических и иных последствий для участников договора. Исполнение договора оплачивается по цене, установленной соглашением сторон.

К договору прилагаются следующие документы, дополняющие и конкретизирующие его содержание:

- протокол соглашения о договорной цене или иной документ, обосновывающий цену договора;
- календарный план-график работ по договору;
- техническое задание на создание АС.

Договор на создание АС или на выполнение отдельных работ жизненного цикла АС имеет локальный характер. Он определяет взаимные обязательства сторон и объект отношений между ними только в установленном этим договором объеме и только на период его действия. Положения договора не могут противоречить законодательству РФ, другим подзаконным и ведомственным актам и документам.

13. Структура и требования ГОСТов к проектной документации разрабатываемой АСОИУ по общесистемным вопросам.

К документам по общесистемным решениям, в общем случае, относят следующие: 1) ведомость эскизного (технического) проекта; 2) пояснительную записку к эскизному (техническому) проекту; 3) схему функциональной структуры; 4) ведомость покупных изделий; 5) описание автоматизируемых функций; 6) описание постановки задачи (комплекса задач); 7) локальную смету и локальный сметный расчет; 8) паспорт; 9) формуляр; 10) проектную оценку надежности системы; 11) общее описание системы; 12) ведомость держателей подлинников; 13) ведомость эксплуатационных документов; 14) программу и методику испытаний (компонентов, комплексов средств автоматизации, подсистем, систем); 15) схему организационной структуры.

Рассмотрим некоторые из этих документов.

1) Документ «Ведомость эскизного (технического) проекта» содержит перечень всех документов, разработанных на соответствующих стадиях создания АС и тех документов, которые были использованы из проектов других АС. Документ выполняется согласно требованиям ГОСТ 2.106.

2) Документ «Пояснительная записка к эскизному (техническому) проекту» содержит четыре раздела: 1) общие положения; 2) описание процесса деятельности; 3) основные технические решения; 4) мероприятия по подготовке объекта автоматизации к вводу системы в действие.

1. В разделе «Общие положения» указывают: 1) наименование проектируемой АС, наименования, номера и даты утверждения документов, на основании которых ведут разработку АС; 2) перечень организаций-исполнителей работ и сроки выполнения стадий; 3) цели, назначение и области использования АС; 4) подтверждение соответствия проектных решений действующим нормам и правилам, например, техники безопасности, пожаро безопасности и т.п.; 5) сведения об использованных нормативно-технических документах, выполненных НИР, изобретениях, использованных при разработке проекта; 6) очередность создания системы и объем каждой очереди.

2. В разделе «Описание процесса деятельности» отражают состав процедур и операций, которые выполняются в процессе функционирования АС, взаимосвязь процессов автоматизированной и неавтоматизированной деятельности, требования к организации работ в условиях АС.

3. Раздел «Основные технические решения» является центральным. Он содержит решения, которые были приняты по следующим вопросам:

- 1) структуре системы и ее подсистем, средствам и способам связи для информационного обмена между компонентами системы и подсистем;
- 2) взаимосвязям АС со смежными системами, обеспечению ее совместимости;

- 3) режимам функционирования, диагностированию работы системы;
- 4) составу функций, комплексов задач и отдельных задач, реализуемых системой (подсистемой);
- 5) составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и т.д.;
- 6) составу программных средств, языкам, алгоритмам процедур и операций и методам их реализации; 7) составу и размещению на объекте комплекса технических средств;
- 8) численности, квалификации и функциям персонала АС, режимам его работы и порядку взаимодействия;
- 9) обеспечению заданных в ТЗ потребительских характеристик системы (подсистемы), определяющих ее качество.

3) Документ «Схема функциональной структуры» – это графическое изображение элементов функциональной структуры АС или ее подсистем, автоматизированные функции и (или) задачи и комплексы задач, а также совокупности действий (операций), выполненных при реализации автоматизированных функций только техническими средствами (автоматически) или только человеком. Для обозначения элементов схемы (ГОСТ 24.302-80), используют прямоугольники с соотношением сторон $b=1.5a$, где $a=20,25,35,40$ мм – высота прямоугольника. Содержание сообщений и (или) сигналов указывают условными обозначениями на связях между элементами схемы. В поясняющих подписях отражают отношения элементов схемы к уровням, объектам, подразделениям управления и т.п. Составляется на стадиях эскизного и технического проектов.

4) Документ «Описание автоматизируемых функций», составляется на стадии «Технический проект» и содержит материалы, касающиеся функциональной части проекта АС. Этот документ содержит следующие разделы: 1) исходные данные; 2) цели АС и автоматизированные функции; 3) характеристика функциональной структуры; 4) типовые решения (если они есть).

В разделе «Исходные данные» приводят перечень исходных материалов и документов, которые были использованы при разработке функциональной части проекта, особенности объекта управления, влияющие на проектные решения по автоматизированным функциям, данные об автоматизированных системах, с которыми связана разрабатываемая, и, наконец, описание информационной модели объекта, вместе с его системой управления.

Раздел «Цели АС и автоматизированные функции» содержит описание тех автоматизированных функций, которые направлены на достижение установленных для объекта целей его деятельности.

Раздел «Характеристика функциональной структуры» содержит перечень подсистем АС с указанием функций и (или) задач, реализуемых в каждой подсистеме, описание процесса выполнения функций, необходимые пояснения к разделению автоматизированных функций на действия, выполняемые техническими средствами и человеком, требования к временному регламенту и характеристикам процесса реализации автоматизированных функций и решения задач.

Раздел «Типовые решения» содержит перечень типовых решений, если они использовались, для автоматизации функций управления.

5) Документ «Описание постановки задачи (комплекса задач)», составляется на стадии «Технический проект» и состоит из трех разделов.

1. В разделе «Характеристики комплекса задач» приводят: 1) назначение комплекса задач; 2) перечень объектов, при управлении которыми решают этот комплекс задач; 3) периодичность и продолжительность решения, условия прекращения решения; 4) связи комплекса задач с другими задачами АС; 5) должности лиц и наименования подразделений, определяющие условия и временные характеристики конкретного решения задачи; 6) распределение действий между персоналом и техническими средствами.
2. Раздел «Выходная информация» содержит: 1) перечень и описание выходных сообщений в виде их идентификатора, формы представления, периодичности и сроков выдачи, получателей и назначения; 2) перечень и описание имеющих самостоятельное смысловое значение структурных единиц выходных сообщений в виде их наименования и идентификатора, требования к точности и надежности вычислений, форме представления (документ, видеокادر, сигнал управления).
3. Раздел «Входная информация» аналогичен предыдущему, но касается входной информации, необходимой для решения задачи или комплекса задач.

6) Документы «Паспорт» и «Формуляр» составляются на стадии «Рабочая документация», по своему содержанию практически идентичны и включают разделы: 1) общие сведения об АС; 2) основные характеристики АС; 3) комплектность АС; 4) свидетельство (акт) о приемке; 5) гарантии изготовителя (разработчика) АС; 6) сведения о рекламациях.

7) Документ «Проектная оценка надежности системы» содержит исходные данные, методику и результаты расчета надежности функционирования АС и ее подсистем. Разрабатывается на стадии «Технический проект» и уточняется на стадии «Рабочая документация».

8) Документ «Общее описание системы» разрабатывается на стадии «Рабочая документация» и предназначен для пользователей разработанной АС, предоставляя им возможность, в достаточно компактной форме, уяснить, как функционирует АС в целом. Документ содержит разделы: 1) назначение системы; 2) описание системы; 3) описание взаимосвязей АС с другими системами; 4) описание подсистем (их структура и функционирование).

9) Документ «Схема организационной структуры» составляется на стадии «Эскизный проект» и затем уточняется на стадии «Технический проект».

Элементы схемы организационной структуры АС или ее подсистем – это условные обозначения структурных подразделений, служб, пунктов управления и отдельных должностных лиц, реализующих функции и задачи управления. Связи на схеме отражают отношения подчиненности при реализации функций управления. При необходимости, на схеме приводят информационные связи между элементами. В поясняющих подписях могут быть указаны функции и задачи управления, реализуемые полностью или частично элементами схемы. Изображения условных обозначений те же, что и в схеме функциональной структуры АС (ГОСТ 24.302-80).

14. Структура и требования ГОСТов к проектной документации разрабатываемой АСОИУ по организационному, техническому и информационному обеспечению.

9.4.1. Документы по организационному обеспечению

Документы по организационному обеспечению АС регламентируют организационную структуру системы управления объектом, в условиях функционирования АС, действия персонала, его функции, права и обязанности. К этим документам относятся: 1) описание организационной структуры; 2) технологическая инструкция; 3) руководство пользователя; 4) описание технологического процесса обработки данных.

Документ «Описание организационной структуры» разрабатывается на стадии «Технический проект» и состоит из трех разделов:

- 1) изменения в организационной структуре управления объектом, где перечисляются подразделения, которые создаются вновь;
- 2) организация новых подразделений, в этом разделе дается описание организационной структуры и функций подразделений, которые создаются с целью обеспечения функционирования АС, описание регламента их работы, перечень категорий работников и число штатных единиц;
- 3) реорганизация существующих подразделений управления, в этом разделе содержатся описания изменений, обусловленных созданием АС, которые необходимо осуществить в каждом из действующих подразделений управления объектом.

Документ «Технологическая инструкция» разрабатывается на стадии «Рабочая документация». В нем приводят сведения о порядке и правилах выполнения операции или комплекса операций технологического процесса обработки данных, перечень должностей персонала, на которые распространяется данная инструкция. Структуру документа определяет разработчик в зависимости от его содержания. Количество таких документов также определяет разработчик.

Документ «Руководство пользователя» составляется на стадии «Рабочая документация». Документ содержит шесть разделов.

1. В разделе «Введение» указывают область применения документа, краткое описание возможностей средства автоматизации и уровень подготовки документов, с которыми необходимо ознакомиться пользователю.

2. Раздел «Назначение и условия применения» содержит информацию о видах деятельности и функции, для автоматизации которых предназначено средство автоматизации, а также о тех условиях, при соблюдении которых обеспечивается применение этого средства.

3. В разделе «Подготовка к работе» указывают состав и содержание дистрибутивного носителя данных, порядок загрузки данных и программ, порядок проверки работоспособности.

4. Раздел «Описание операций» содержит описание всех выполняемых функций, задач, комплексов задач и процедур; описание операций технологического процесса обработки данных, необходимых для выполнения этих функций, задач и процедур; необходимые подготовительные, основные и заключительные действия, в требуемой последовательности; ресурсы, расходуемые на операции.

5. В разделе «Аварийные ситуации» указывают действия, при отказе технических средств, по восстановлению программ и данных, действия в случаях обнаружения несанкционированного вмешательства в данные, при аварийных ситуациях.

6. Раздел «Рекомендации по освоению» содержит рекомендации по освоению и эксплуатации, в том числе описание контрольного примера, правила его запуска и выполнения.

Количество таких документов определяет разработчик.

Документ «Описание технологического процесса обработки данных (включая телеобработку)» составляют на стадии «Рабочая документация». Этот документ содержит два раздела.

1. В разделе «Технологический процесс сбора и обработки данных на периферийных устройствах, при децентрализованной обработке данных» указывают состав и последовательность выполнения технологических операций по сбору, регистрации, подготовке, контролю, передаче, обработке и отображению информации, а также перечень документов, сопровождающих каждую операцию в данном технологическом процессе.

2. Раздел «Технологический процесс обработки данных на вычислительном центре» содержит состав и последовательность выполнения технологических операций по приему, контролю, обработке, хранению, выдаче данных и других операций, выполняемых на вычислительном центре, а также перечень документов, сопровождающих данный технологический процесс.

9.4.2. Документы с решениями по техническому обеспечению

Документы по техническому обеспечению АС регламентируют вопросы построения структуры комплекса технических средств АС, расположения этих средств на объекте управления. В них также дается принципиальная схема соединений технических средств АС, план расположения этих средств, чертеж общего вида, инструкции по эксплуатации комплекса технических средств, другая информация, касающаяся вопросов технической реализации проектных решений по АС.

Наибольший интерес представляют следующие документы:

- 1) «Описание комплекса технических средств (КТС)»
- 2) «Схема структурная комплекса технических средств».

Документ «Описание комплекса технических средств» разрабатывается на стадии «Технический проект». Этот документ содержит четыре раздела.

В разделе «Общие положения» приводят исходные данные, которые были использованы при проектировании технического обеспечения АС.

В разделе «Структура комплекса технических средств» дают обоснование выбора структуры комплекса технических средств; описание его функционирования, в том числе в пусковых и аварийных режимах, приводят описание размещения КТС на объекте, обоснование применения и технические требования к оборудованию, дают обоснование методов защиты технических средств от различного вида воздействий, приводят результаты проектной оценки надежности КТС.

В разделе «Средства вычислительной техники» дается обоснование и описание основных решений по выбору ЭВМ и периферийных технических средств, приводится обоснование численности персонала, обеспечивающего

функционирование технических средств, даются технические решения по оснащению рабочих мест персонала, приводится описание особенностей функционирования технических средств в пусковом, нормальном и аварийном режимах.

В разделе «Аппаратура передачи данных» дается обоснование и описание решений по выбору телеобработки и передачи данных, технических средств сопряжения с каналами связи, приводятся требования к арендуемым каналам связи, даются сведения о размещении абонентов и объемно-временных характеристиках передаваемых данных, приводятся показатели надежности, достоверности и других технических характеристик средств телеобработки и передачи данных.

Документ «Схема структурная комплекса технических средств» разрабатывается на стадии «Эскизный проект» и затем уточняется на стадии «Технический проект». Документ содержит состав комплекса технических средств и связи между этими техническими средствами или группами технических средств. Элементы схемы структурной КТС – это условные обозначения отдельных технических средств или их групп. На схеме отражают информационные связи между элементами. В поясняющих надписях могут быть отражены эти сопряжения, т.е. ранг сопряжения, вид носителя, код устройства и т.д.

Структура КТС АС может быть представлена несколькими схемами, первая из которых отражает укрупненную схему в целом. При изображении схемы используют условные обозначения согласно ГОСТ 24.303-80.

Наименования устройств (или их комплексов), изображения и размеры их общих обозначений приведены в табл. 9.1 (ГОСТ 24.303-80), где $a=10,15,25$ мм или кратное 5, $b=1.5a$.

Для изображения комплекса устройств внутри общего обозначения комплекса (символ 1, табл. 9.1), помещают символ устройства того же назначения. Для изображения устройства управления внутри символа соответствующего устройства помещают общее обозначение устройства управления. Взаимное расположение символов друг в друге не регламентировано.

i. Документы с решениями по информационному обеспечению

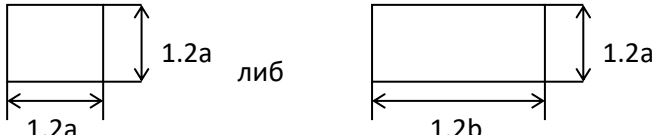

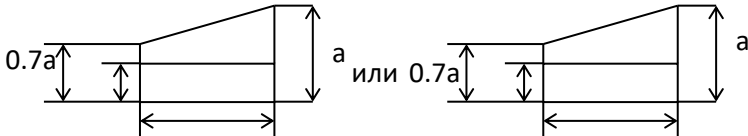
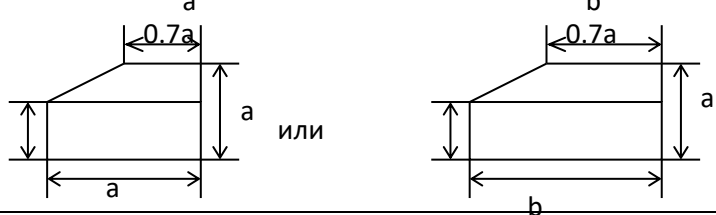
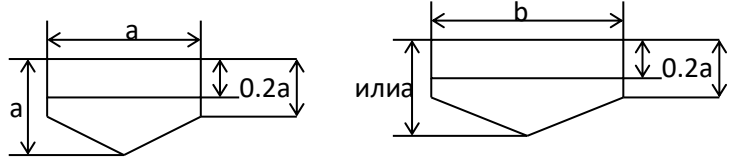
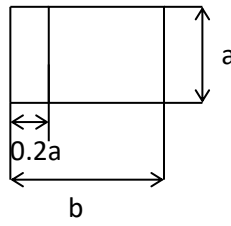
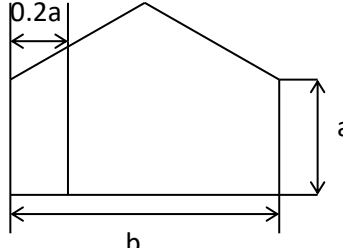
Описание проектных решений по информационному обеспечению АСОИУ содержится в следующих двенадцати документах: 1) «Перечень входных (выходных) сигналов»; 2) «Описание информационного обеспечения системы»; 3) «Ведомость машинных носителей информации»; 4) «Описание организации информационной базы»; 5) «Описание систем классификации и кодирования»; 6) «Описание массива информации»; 7) «Чертеж формы документа (видео кадра)»; 8) «Массив входных данных»; 9) «Каталог базы данных»; 10) «Состав выходных данных (сообщений)»; 11) «Инструкция по формированию и ведению базы данных (набора данных)».

Документ «Перечень входных сигналов и данных» разрабатывается на стадии «Технический проект». Этот документ содержит два раздела.

1. Раздел «Перечень входных сигналов» содержит: а) следующую информацию о каждом аналоговом сигнале: наименование и единицу измерения измеряемой величины, диапазон изменения, требования к точности и периодичности измерения, тип сигнала; б) для каждого

Таблица 9.1. Наименования устройств, изображения и размеры их общих обозначений

Наименование устройства	Изображение и размеры
-------------------------	-----------------------

1. Комплекс устройств (общее обозначение)	
2. Устройство (общее обозначение)	
3. Устройство терминальное	
4. Устройство регистрации первичной обработки информации	
5. Устройство подготовки данных	
6. ЦВМ общего назначения	
7. Специализированная вычислительная машина	

дискретного сигнала дается следующая информация: наименование, разрядность и периодичность, тип сигнала; в) для каждого сигнала типа «да - нет» указывается источник формирования информации.

2. Раздел «Перечень входных данных» содержит наименования, кодовые обозначения и значности реквизитов всех входных данных с указанием наименований и кодовых обозначений документов или сообщений, содержащих эти данные.

Документ «Перечень выходных сигналов (документов)» состоит из двух разделов.

1. В разделе «Перечень выходных сигналов» приводится перечень: выходных сигналов с указанием их наименований, назначения, единицы измерения, диапазона изменения, способа представления; пользователей информации.

2. Раздел «Перечень выходных документов» содержит список выходных документов с указанием их наименований, кодовых обозначений, перечня и значности реквизитов, пользователей информации. Разрабатывается данный документ на стадии «Технический проект».

Документ «Описание информационного обеспечения системы» составляют на стадии «Технический проект». Он состоит из шести разделов.

1. В разделе «Состав информационного обеспечения» указывают наименование и назначение всех баз данных и наборов данных.

2. Раздел «Организация информационного обеспечения» содержит принципы организации информационного обеспечения системы, обоснование выбора носителей данных и принципы распределения информации по этим носителям, описание принятых видов и методов контроля в маршрутах обработки данных при создании и функционировании внешнимашинной и внутримашинной информационных баз, описание решений, обеспечивающих информационную совместимость АС с другими автоматизированными системами.

3. В разделе «Организация сбора и передачи информации» приводят перечень источников и носителей информации с указанием оценки интенсивности и объема потоков информации, а также дают описание общих требований к организации сбора, передачи, контроля и корректировки информации.

4. Раздел «Построение системы классификации и кодирования» содержит: 1) описание принятых для применения в АС методов классификации объектов во вновь разработанных классификаторах и в тех действующих классификаторах, из которых используется часть кода; 2) методы кодирования объектов классификации во вновь разработанных классификаторах.

5. В разделе «Организация внутримашинной информационной базы» дают описание принципов построения внутримашинной информационной базы и характеристики ее состава и объема. Здесь же дают описание информационной базы на уровне баз данных с описанием характера взаимосвязей баз данных и указанием функций АС, при реализации которых используют каждую базу данных, а также характеристики данных, содержащихся в каждой базе данных.

6. В разделе «Организация внешнимашинной информационной базы» приводят характеристики состава и объема внешнимашинной информационной базы, принципы ее построения.

Документ «Ведомость машинных носителей информации» составляют на стадии «Рабочая документация». Он содержит обозначения и наименования документов, выполненных на машинных носителях.

Документ «Описание организации информационной базы» разрабатывают на стадии «Технический проект». Документ состоит из двух частей: 1) описание внутримашинной информационной базы; 2) описание внешнимашинной информационной базы.

Части имеют одну и ту же структуру и содержат следующие разделы: 1) логическая структура; 2) физическая структура (для внутримашинной информационной базы); 3) организация ведения информационной базы.

1. В разделе «Логическая структура» приводят описание состава данных, их форматов и взаимосвязей между данными.

2. Раздел «Физическая структура» содержит описание выбранного варианта расположения данных на конкретных машинных носителях.

При описании структуры внутримашинной информационной базы приводят перечни баз данных и массивов и логические связи между ними. Для массива информации указывают логическую структуру внутри массива или дают ссылку на документ «Описание массива информации».

При описании структуры внешнимашинной информационной базы приводят перечень документов и других информационных сообщений, использование которых предусмотрено в системе, с указанием автоматизируемых функций, при реализации которых формируют и используют данный документ.

3. В разделе «Организация ведения информационной базы» при описании внутримашинной базы, приводят последовательность процедур при создании и обслуживании базы с указанием регламента выполнения процедур и средств защиты базы от разрушений и несанкционированного доступа, а также с указанием связей между массивами баз данных и массивами входной информации.

При описании внешнимашинной информационной базы приводят последовательность процедур по маршруту движения групп документов до передачи их на ВЦ, а также маршрут движения выходных документов.

Документ «Описание систем классификации и кодирования» разрабатывается на стадии «Технический проект». Он содержит перечень применяемых зарегистрированных классификаторов, описание метода кодирования, структуры и длины кода, указания к системе классификации, другие сведения по усмотрению разработчика.

Документ «Описание массива информации» содержит наименование и обозначение массива; наименование носителей информации; перечень реквизитов в порядке их следования в записях массива с указанием их количественных характеристик; оценку объема массива. Составляют документ на стадии «Технический проект».

Документ «Массив входных данных» составляется на стадии «Рабочая документация». Этот документ содержит перечень входных данных с указанием их наименований, кодовых обозначений и значности реквизитов, наименований и кодовых обозначений документов или сообщений, содержащих данные.

Документ «Каталог базы данных» составляют на стадии «Рабочая документация» и содержит перечень объектов предметной области АС, информация о которых включена в базу данных.

Документ «Состав выходных данных (сообщений)» также составляют на стадии «Рабочая документация». Он содержит перечень выходных данных с указанием их наименований, кодовых обозначений и значности реквизитов, а также наименований и кодовых обозначений документов и сообщений, содержащих эти данные.

Документ «Инструкция по формированию и ведению базы данных (набора данных)» разрабатывается на стадии «Рабочая документация». Он содержит четыре раздела: 1) правила подготовки данных; 2) порядок и средства заполнения базы данных; 3) процедуры изменения и контроля базы данных; 4) порядок и средства восстановления базы данных.

1. В разделе «Правила подготовки данных» приводят порядок отбора информации, правила подготовки и кодирования информации, формы ее представления и правила заполнения этих форм, порядок внесения изменений информации.

2. Раздел «Порядок и средства заполнения базы данных» содержит состав технических средств, правила, порядок, последовательность и описание процедур, которые используются при заполнении базы данных.

3. В разделе «Процедуры изменения и контроля базы данных» приводят состав и последовательность выполнения процедур по контролю и изменению содержания базы данных.

4. В разделе «Порядок и средства восстановления базы данных» приводят описание средств защиты базы данных от разрушения и несанкционированного доступа, правила, средства и порядок проведения процедур по копированию и восстановлению базы данных.

15. Структура и требования ГОСТов к проектной документации разрабатываемой АСОИУ по программному обеспечению.

9.4.4. Документы с решениями по программному обеспечению

Состав документов по программному обеспечению АС определяют ГОСТ 34.201-89 (Виды, комплектность и обозначение документов при создании автоматизированных систем) и ГОСТ 19.101-77, регламентирующий виды документов на программные средства, используемые при создании АС.

Документы по программному обеспечению АС можно разделить на четыре группы:

- 1) для описания проектных решений по программному обеспечению;
 - 2) для установления требований к программе или комплексу программ;
 - 3) для описания решений, обеспечивающих сопровождение, изготовление и эксплуатацию программы или комплекса программ;
 - 4) для проверки работоспособности программы или комплекса программ
- (см. рис 9.4.1).

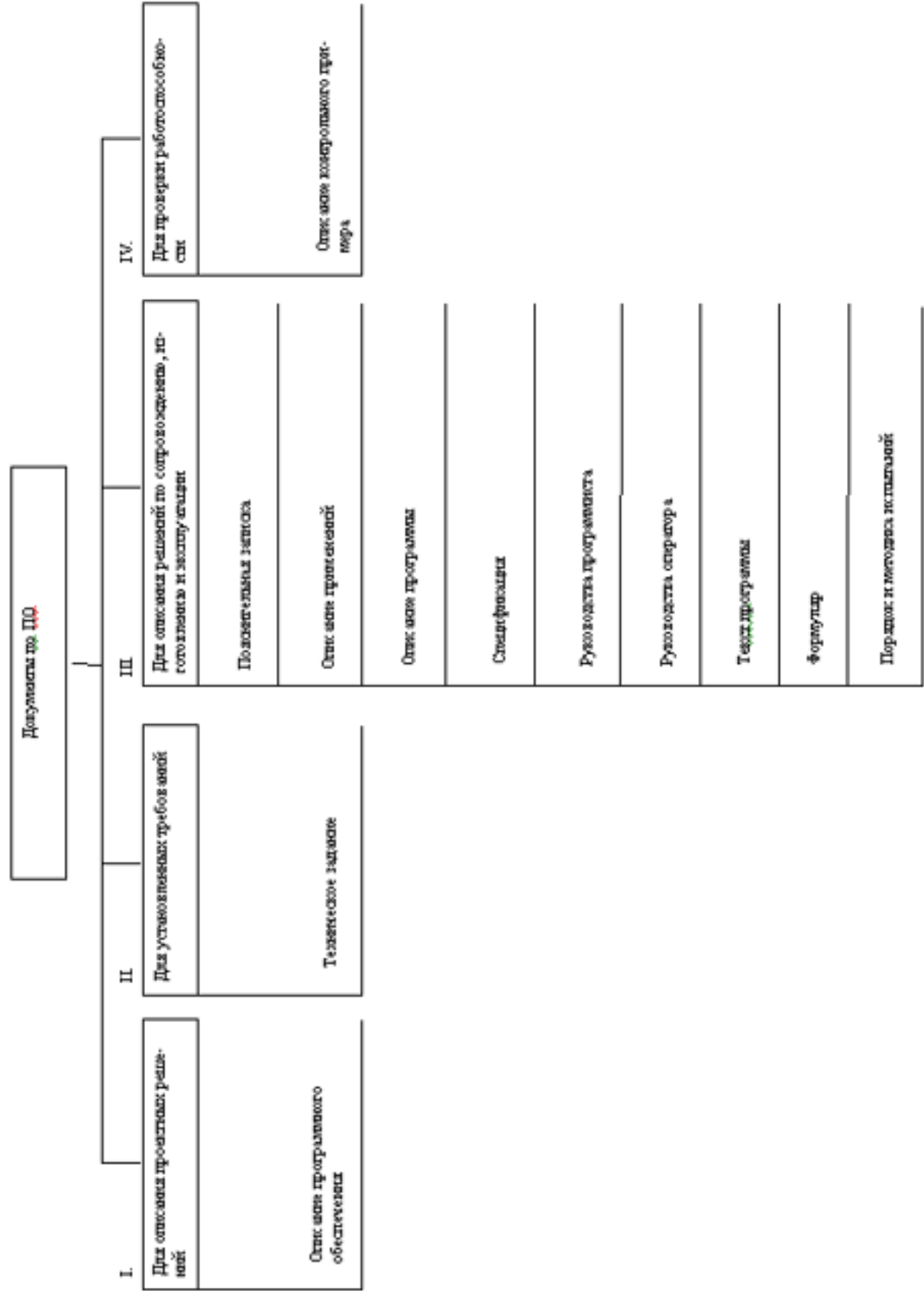


Рис. 9.4.1.

I. К первой группе относится документ «Описание программного обеспечения». Этот документ, согласно ГОСТ 34.201-89, разрабатывается на стадии «Технический проект». Его состав определен РД 50-34.698-90. Документ состоит из следующих шести разделов:

1. «Вводная часть», где содержатся основные сведения об АС, необходимые для разработки программного обеспечения;
2. «Структура программного обеспечения», где приводится перечень частей программного обеспечения с указанием взаимосвязей и обоснованием выделения каждой из них;
3. «Функции частей программного обеспечения», этот раздел состоит из подразделов, в каждом из которых приводится назначение и описание основных функций соответствующей части;
4. «Методы и средства разработки программного обеспечения», где дается перечень методов программирования и средств разработки программного обеспечения АС с указанием частей программного обеспечения, при разработке которых использовались эти методы и средства;
5. «Операционная система», где дается обоснование выбранной операционной системы, наименование руководства, в соответствии с которым должна осуществляться генерация выбранного варианта операционной системы; требования к варианту генерации выбранной версии операционной системы;
6. «Средства, расширяющие возможности операционной системы», этот раздел состоит из подразделов, в которых, для каждого используемого средства, расширяющего возможности операционной системы, указывается: наименование, обозначение и краткая характеристика средства с обоснованием необходимости его применения и указанием источника, где дано подробное описание выбранного средства; наименование руководства, в соответствии с которым следует настраивать используемое средство на конкретное применение; требования к настройке используемого средства.

II. Вторая группа состоит из одного документа, определяющего требования к программе или комплексу программ, - «Техническое задание».

Этот документ составляют в соответствии с ГОСТ 19.201-78 (ЕСПД. Техническое задание. Требования к содержанию и оформлению).

«Техническое задание» содержит следующие разделы: 1) введение; 2) основание для разработки; 3) назначение разработки; 4) требования к программной документации; 5) технико-экономические показатели; 6) стадии и этапы разработки; 7) порядок контроля и приемки.

Допускается вводить новые разделы или объединять отдельные из них.

Наиболее важным в этом документе является раздел, касающийся требований к программе. Этот раздел должен включать следующие подразделы: 1) требования к функциональным характеристикам (т.е. требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.д.); 2) требования к надежности (т.е. требования к обеспечению устойчивого функционирования, контролю входной и выходной информации, времени восстановления после отказа и т.д.); 3) требования к эксплуатации, в том числе составу и параметрам технических средств; 4) требования к информационной и программной совместимости; 5) требования к маркировке, упаковке, транспортировке и хранению, другие требования.

III. Третья группа документов по ПО посвящена описанию проектных решений по сопровождению, изготовлению и эксплуатации программы или комплекса программ. К этой группе относят документы: 1) «Пояснительная записка»; 2) «Описание применения»; 3) «Описание программы»; 4) «Спецификации»; 5) «Руководство программиста»; 6) «Руководства оператора»; 7) «Текст программы»; 8) «Формуляр»; 9) «Порядок и методика испытаний». Рассмотрим наиболее важные из них.

III.1. Документ «Пояснительная записка» составляется в соответствии с ГОСТ 19.404-79. Он содержит следующие разделы: 1) введение; 2) назначение и область применения; 3) технические характеристики; 4) ожидаемые технико-экономические показатели; 5) источники, использованные при разработке. Наиболее объемным является раздел «Технические характеристики», в котором содержатся следующие подразделы: 1) постановка задачи, описание применяемых математических методов, допущений и ограничений, связанных с выбранным математическим аппаратом; 2) описание алгоритма и (или) функционирования программы с обоснованием выбора схемы алгоритма, возможные взаимодействия программы с другими программами; 3) описание и обоснование выбора метода организации входных и выходных данных; 4) описание и обоснование выбора состава технических и программных средств, распределение носителей данных, которые использует программа.

III.2. Документ “Описание применения” составляется в соответствии с ГОСТ 19.502-78. Он содержит разделы: назначение программы; условия применения; описание задачи; входные и выходные данные.

- 1) В разделе “Назначение программы” указывают назначение и возможности программы, ее основные характеристики и ограничения, накладываемые на область применения.
- 2) В разделе “Условия применения” приводят требования к необходимым для данной программы техническим средствам и другим программам, общие характеристики входной и выходной информации, требования и условия организационного, технического, технологического и др. характера.
- 3) В разделе “Описание задачи” указывают определение задачи и методы ее решения.
- 4) В разделе “Входные и выходные данные” приводят сведения о входных и выходных данных.

III.3. Документ “Описание программы” составляется с требованиями ГОСТ 19.402-78. Если программа или комплекс программ получена за счет использования ранее разработанных программных средств, то этот документ дополняют разделом “Настройка программных средств”.

Согласно ГОСТ 19.402-78, документ “Описание программы” должен содержать следующие семь разделов: 1) общие сведения; 2) функциональное назначение; 3) описание логической структуры; 4) используемые технические средства; 5) вызов и загрузка; 6) входные данные; 7) выходные данные. В зависимости от особенностей разработанной программы, допускается вводить дополнительные разделы, либо объединять отдельные разделы.

- 1) В разделе “Общие сведения” указывают обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы, языки программирования.
- 2) В разделе “Функциональное назначение” приводят классы решаемых задач, назначение программы и сведения о функциональных ограничениях по применению.
- 3) Раздел “Описание логической структуры” содержит: алгоритм программы; используемые методы; структуру программы с указанием связей между ее элементами; связи программы с другими программами. Этот раздел выполняют с учетом текста программы на исходном языке.
- 4) Раздел “Используемые технические средства” содержит сведения о типе ЭВМ или других устройств.
- 5) В разделе “Вызов и загрузка” указывают способ вызова программы с соответствующего носителя данных, входные точки в программу. Возможно указание адресов загрузки, сведений об использовании оперативной памяти, объеме программы.
- 6)– 7) В разделах “Входные данные” “Выходные данные” указывают характер, организацию, формат, описание и способ кодирования входных (выходных) данных. Для входных данных еще указывают их предварительную подготовку, если она есть.

III.4. Документ “Руководство программиста” по составу разделов и их содержанию, должен соответствовать ГОСТ 19.504-79. Он содержит следующие разделы: 1) назначение и условия применения программы; 2) характеристики программы; 3) обращение к программе; 4) входные и выходные данные; 5) сообщения; 6) сведения о форме представления программы. В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

- 1) В разделе “Назначение и условия применения программы” указывают назначение и функции, выполняемые программой, условия ее выполнения (объем оперативной памяти, требования к составу и параметрам периферийных устройств, программному обеспечению и т.п.).
- 2) В разделе “характеристики программы” приводят описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и т.п.).
- 3) В разделе “Обращение к программе” приводят описание процедур вызова программы (способы передачи управления и параметров данных).
- 4) В разделе “Входные и выходные данные” приводят описание организации используемой входной и выходной информации и, при необходимости, ее кодирования.
- 5) В разделе “Сообщения” указывают тексты сообщений, выдаваемых пользователю в ходе выполнения программы, описание их содержания и действия, которые необходимо предпринимать по этим сообщениям.

- б) Кроме указанных разделов документ “Руководство программиста” должен включать раздел “Сведения о форме представления программы (комплекса программ)”. В этом разделе даются сведения о носителе, на котором записана программа, о содержании и системе кодирования информации, записанной на носителе, сведения, необходимые для чтения информации с носителя.

Если программа (комплекс программ) допускает настройку на условия конкретного применения, то в документ “Руководство программиста” дополнительно включают разделы: структура программы; настройка программы; дополнительные возможности; сообщения программному программисту. Содержания этих разделов должно соответствовать требованиям ГОСТ 19.503-79 (Руководство системного программиста. ЕСПД). Согласно этому ГОСТу в разделе “Структура программы” приводятся сведения о структуре программы, ее составных частях и связях между ними и с другими программами.

В разделе “Настройка программы” приводят описание действий по настройке программы на условия конкретного применения (настройка на состав технических средств, выбор функций и др.).

В разделе “Дополнительные возможности” дают описание дополнительных разделов функциональных возможностей программы и способов их выбора.

В разделе “Сообщения системному программисту” приводят тексты сообщений, выдаваемых в ходе выполнения настройки и проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые должен предпринять системный программист при их появлении.

Требования к документам “Спецификация”, “Руководство оператора”, “Текст программы”, “Формуляр”, “Порядок и методика испытаний” приведены в ГОСТ 19.202-78, ГОСТ 19.505-79, ГОСТ 19.401-78, ГОСТ 19.501-78, ГОСТ 19.301-79.

IV. Четвертую группу документов по ПО составляют документы, содержащие проектные решения, обеспечивающие проверку работоспособности программы или комплекса программ, содержатся в документе «Описание контрольного примера» ГОСТ 24.207-80. Этот документ состоит из следующих разделов:

- 1) «Назначение», где содержится перечень параметров и краткая характеристика функций из числа реализуемых программой, которые проверяются контрольным примером;
- 2) «Исходные данные», в котором приводится описание исходных данных для проверки программы и сами исходные данные;
- 3) «Результаты расчета», в котором содержатся результаты обработки исходных данных проверяемой программой, позволяющие оценить правильность выполнения этой программой ее функций;
- 4) «Проверка программы», где должны быть приведены описания: состава технических средств, необходимых для выполнения программы; процедур формирования исходных данных для проверки программы, ее вызова и получения исходных данных; действий оператора при подготовке исходных данных и проверке программы на контрольном примере.

16. Требования ГОСТов к выполнению схем алгоритмов, программ, данных и систем.

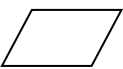
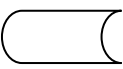


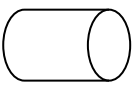

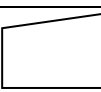
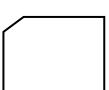

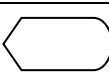

Требования к выполнению схем алгоритмов, программ, данных и систем установлены ГОСТ 19.701-90 (ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения). В соответствии с этим документом кроме схем алгоритмов составляют:



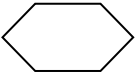
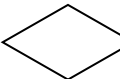
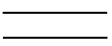
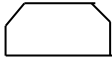
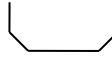



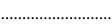



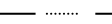
- 1) схемы данных;
- 2) схемы программ;
- 3) схемы работы системы;
- 4) схемы взаимодействия программ;
- 5) схемы ресурсов системы.

Для изображения всех указанных выше схем, включая схемы алгоритмов, используют символы. Эти символы сведены в четыре группы: 1) символы данных; 2) символы процесса; 3) символы линий; 4) специальные символы. Изображение символов, их назначение и возможности применения для изображения перечисленных выше схем, приведено в таблице 9.4.4.

Примеры фрагментов схем с использованием символов: «Параллельные вычисления»; «Границы цикла»; «Пунктирная линия», приведены на рис. 9.4.2, 9.4.3, 9.4.4, 9.4.5.

Таблица 9.4.4. Применение символов

Символ	Наименование символа	Схе ма дан ных	Схем а прогр амм	Схем а боты систе мы	Схема Взаим одейст вия програ мм	Схем а ресур сов систе мы
	I. Символы данных					
	Основные					
	Данные (носитель данных не определен)	+	+	+	+	+
	Запоминаемые данные (носитель данных не определен)	+	-	+	+	+
	Специфические					
	Оперативное запоминающее устройство	+	-	+	+	+
	Запоминающее устройство с последовательной выборкой (магнитная лента, кассета с магнитной лентой)	+	-	+	+	+
	Запоминающее устройство с прямым доступом (магнитный диск, магнитный барабан, гибкий магнитный диск)	+	-	+	+	+
	Документ (машинограмма, документ для оптического или магнитного считывания, микрофильм, рулон ленты с итоговыми данными, бланки ввода данных)	+	-	+	+	+
	Ручной ввод данных с клавиатуры, световым пером, кнопками и т.д.	+	-	+	+	+
	Карта (данные на перфоленте, магнитной карте, карте со сканируемыми метками и т.п.)	+	-	+	+	+
	Бумажная лента	+	-	+	+	+
	Дисплей (данные на экране для визуального наблюдения)	+	-	+	+	+
	II. Символы процесса					
	Основные					
	Процесс (функция обработки данных любого вида)	+	+	+	+	+
	Специфические					

	Предопределенный процесс	-	+	+	+	-
	Ручная операция	+	-	+	+	-
	Подготовка (отображает модификацию команды или группы команд с целью воздействия на последующую функцию: установка переключателя, инициализация программы и т.п.)	+	+	+	+	-
	Решение (отображает решение или функцию типа переключателя с одним входом и несколькими альтернативными выходами)	-	+	+	-	-
	Параллельные действия для отображения синхронизации двух или более параллельных операций	-	+	+	+	-
	Граница цикла – начало	-	+	+	-	-
	Граница цикла – окончание	-	+	+	-	-
	III. Символы линий.					
	Основные					
	Линия	+	+	+	+	+
	Специфические					
	Передача управления	-	-	-	+	-
	Канал связи	+	-	+	+	+
	Пунктирная линия (отражает альтернативную связь или используется для обведения участков схемы)	+	+	+	+	+
	IV. Специальные символы					
	Соединитель (отражает вход и выход в часть системы)	+	+	+	+	+
	Терминатор (отражает начало или конец схемы)	+	+	+	-	-
	Комментарий в схеме	+	+	+	+	+
	Пропуск в схеме	+	+	+	+	+

Примечание. Знак «+» указывает, что символ используют в данной схеме, знак «-» - не используют.

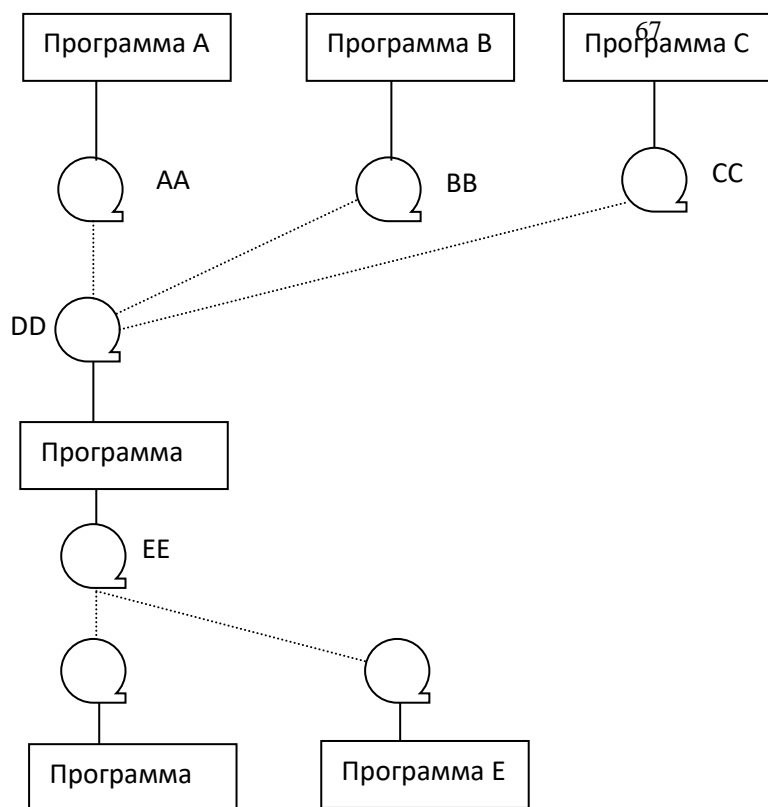
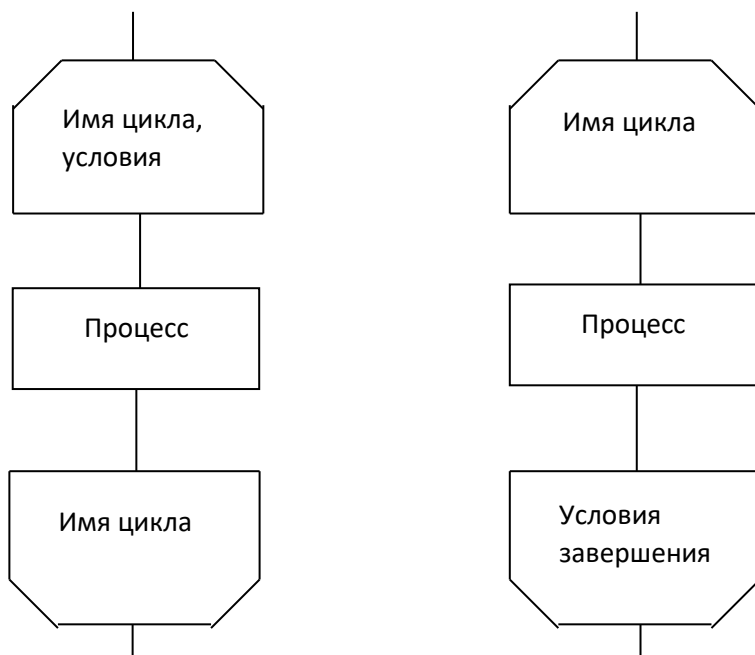


Рис.9.4.2. Пример изображения альтернативных входов и выходов из запоминающих устройств с последовательным доступом



аннотированного



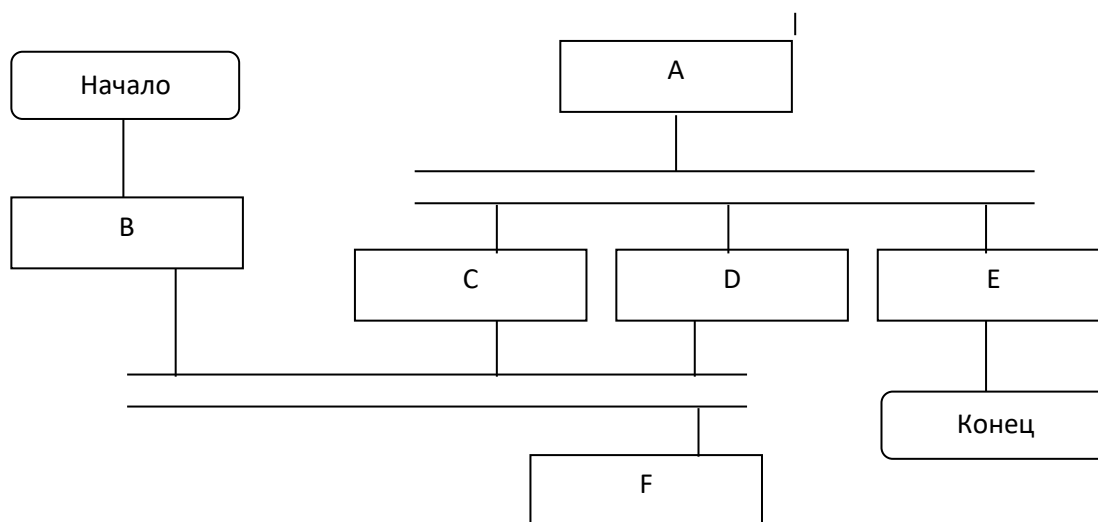
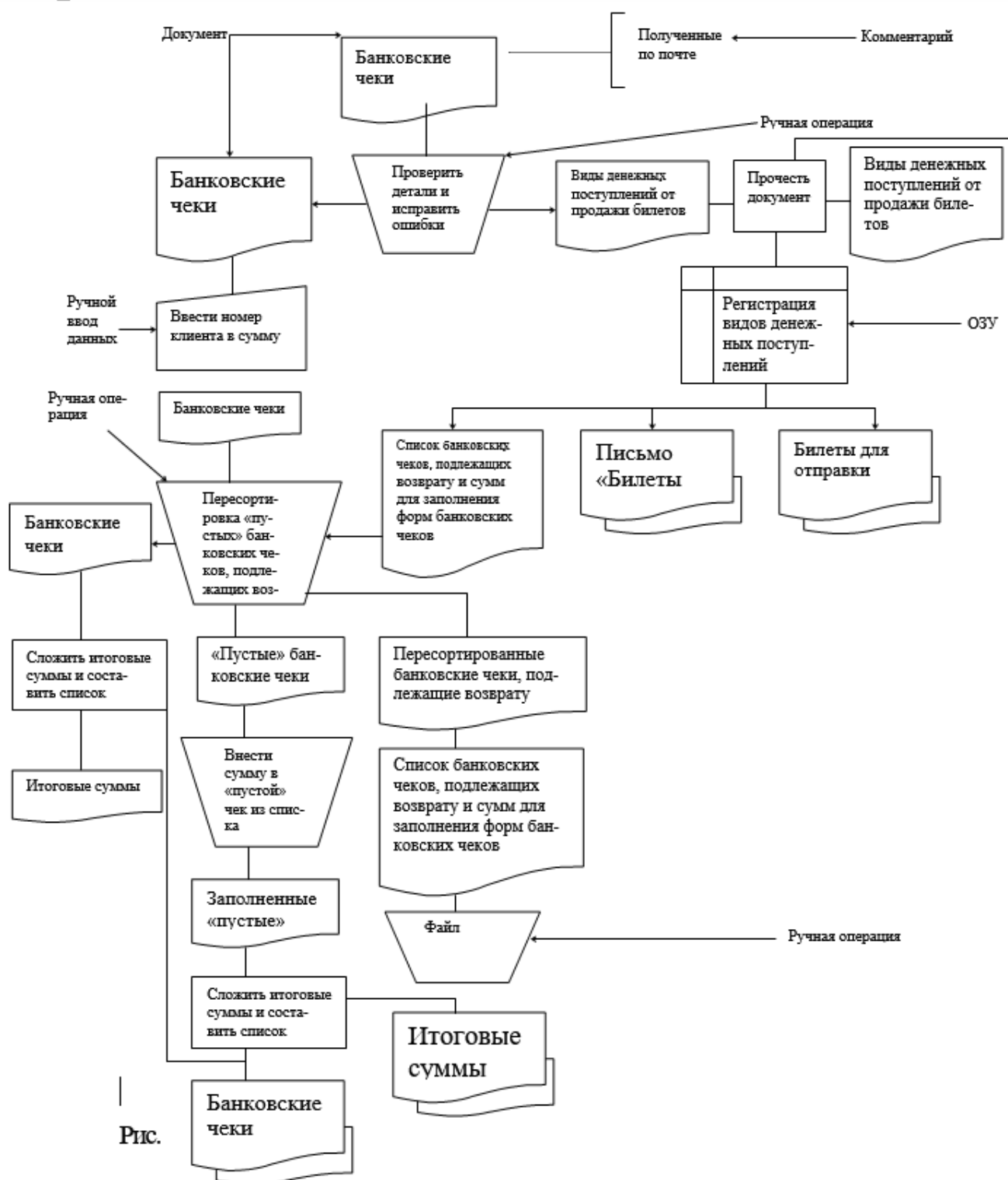


Рис.9.4.5. Пример использования символа «Параллельные действия». (Процессы C, D, E могут начинаться одновременно после выполнения процесса A, процесс F может начинаться после выполнения процессов B, C, D, процесс C может завершиться раньше или позже процесса D).

Рассмотрим, что обозначает каждая из перечисленных выше схем:

- 1) схемы данных;
- 2) схемы программ;
- 3) схемы работы системы;
- 4) схемы взаимодействия программ;
- 5) схемы ресурсов системы.

Схема данных отображает путь данных при решении задач и определяет этапы их обработки, а также различные применяемые носители данных. Схема данных состоит из символов данных, символов процесса над данными, символов линий для указания потоков данных между процессами и (или) носителями данных, специальных символов для обеспечения написания и чтения схемы. Пример схемы данных приведен на рис. 9.4.6.



9.4.6. Пример схемы данных

Схема программ отображает последовательность операций в программе. Схема программы состоит из символов процесса, указывающих фактически операции обработки данных, линейных символов для указания потока управления и специальных символов для облегчения написания и чтения программы. Примеры схем программ приведены на рис. 9.4.7 и 9.4.8

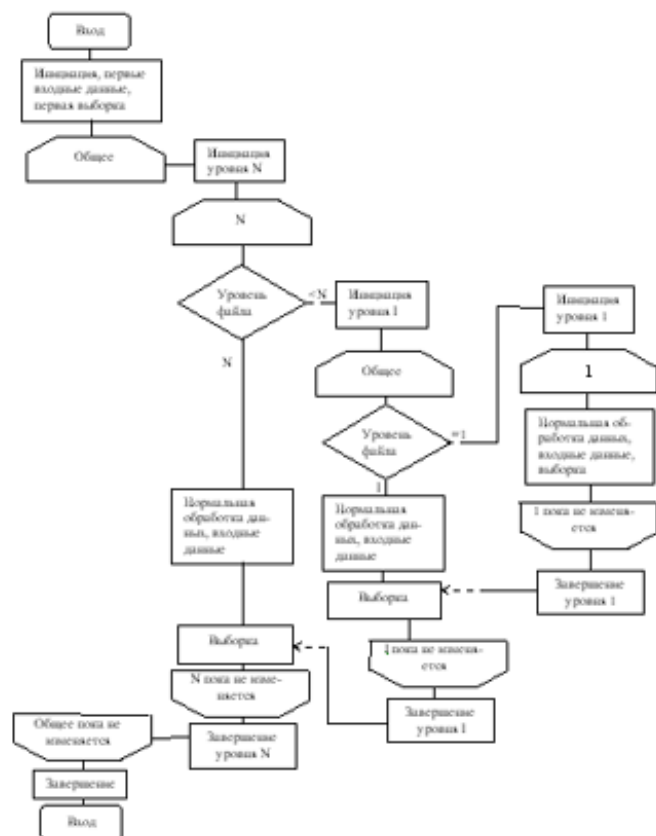


Схема работы системы отображает управление операциями и поток данных в системе. Схема работы системы состоит из: 1) символов данных, указывающих на наличие данных и (или) вид носителя данных; 2) символов процесса, указывающих операции, которые следует выполнить над данными, а также определяющих логический путь, которого следует придерживаться; 3) линейных символов, указывающих потоки данных между процессами и (или) носителями данных, а также поток управления между процессами; 4) специальных символов, используемых для облегчения написания и чтения схемы. Пример схемы работы системы приведен на рис. 9.4.9.

Схема взаимодействия программ отображает путь активаций программ и взаимодействий с соответствующими данными. Каждая программа в такой схеме показывается только один раз (в отличие от схемы работы системы, где программа может изображаться более чем в одном потоке управления). Схема взаимодействия программ состоит из следующих элементов: 1) символов данных, указывающих на наличие данных; 2) символов процесса, указывающих на операции, которые следует выполнить над данными; 3) линейных символов, отображающих поток между процессами и данными, а также инициации процессов; 4) специальных символов, используемых для облегчения написания и чтения схемы. Пример схемы приведен на рис. 9.4.10.

Схема ресурсов системы отображает конфигурацию блоков данных и обрабатывающих блоков, которая требуется для решения задачи или набора задач. Схема ресурсов системы состоит из следующих элементов: 1) символы данных, отображающих входные, выходные и запоминающие устройства ЭВМ; 2) символы процесса, отображающих процессоры (центральные процессоры, каналы и т.д.); 3) линейные символы, отображающие передачу данных между устройствами ввода/вывода и процессорами, а также передачу управления между процессорами; 4) специальные символы для облегчения написания и чтения схемы. Пример схемы ресурсов системы приведен на рис. 9.4.11.

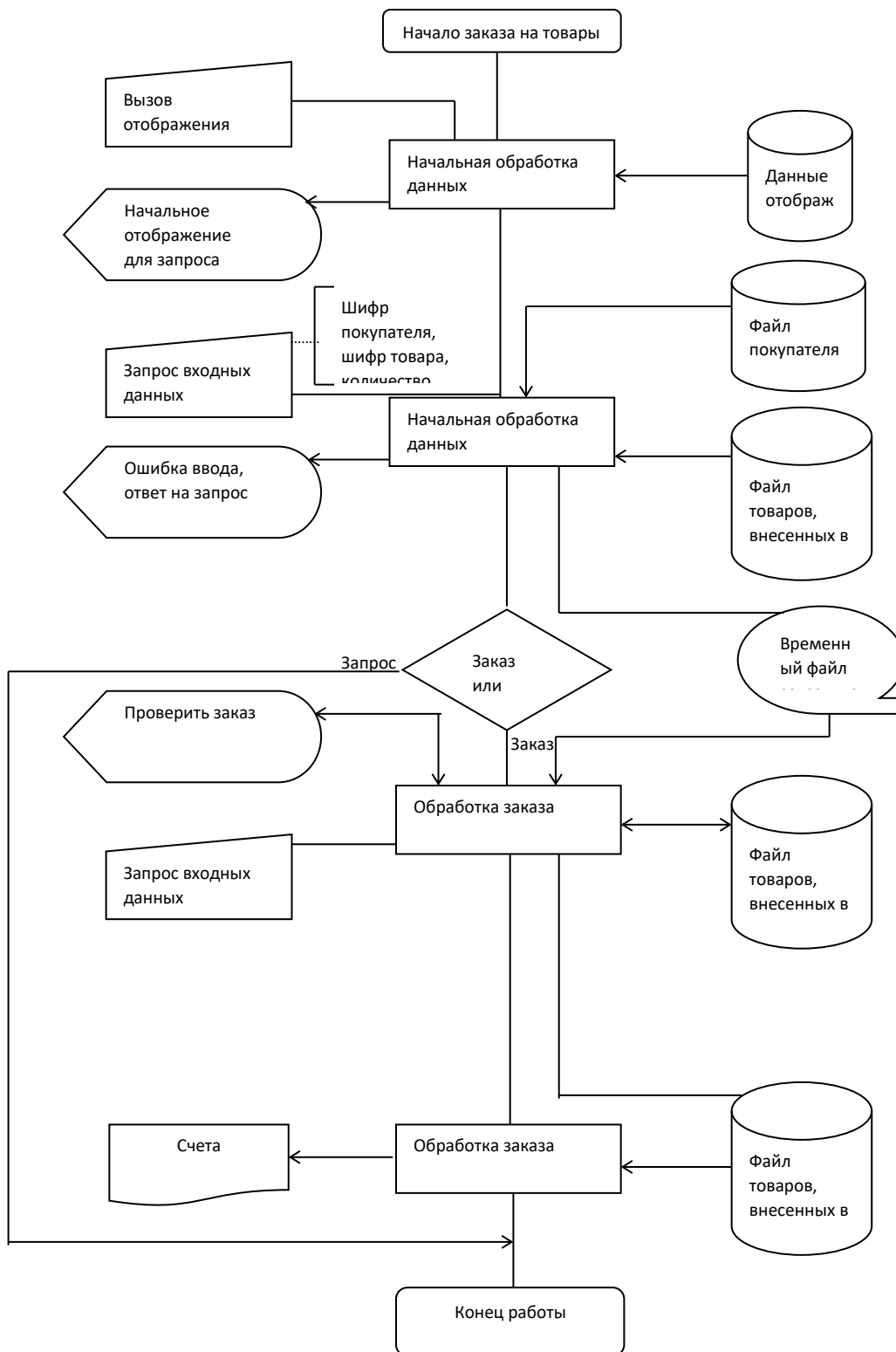


Рис. 9.4.9. Пример схемы работы системы

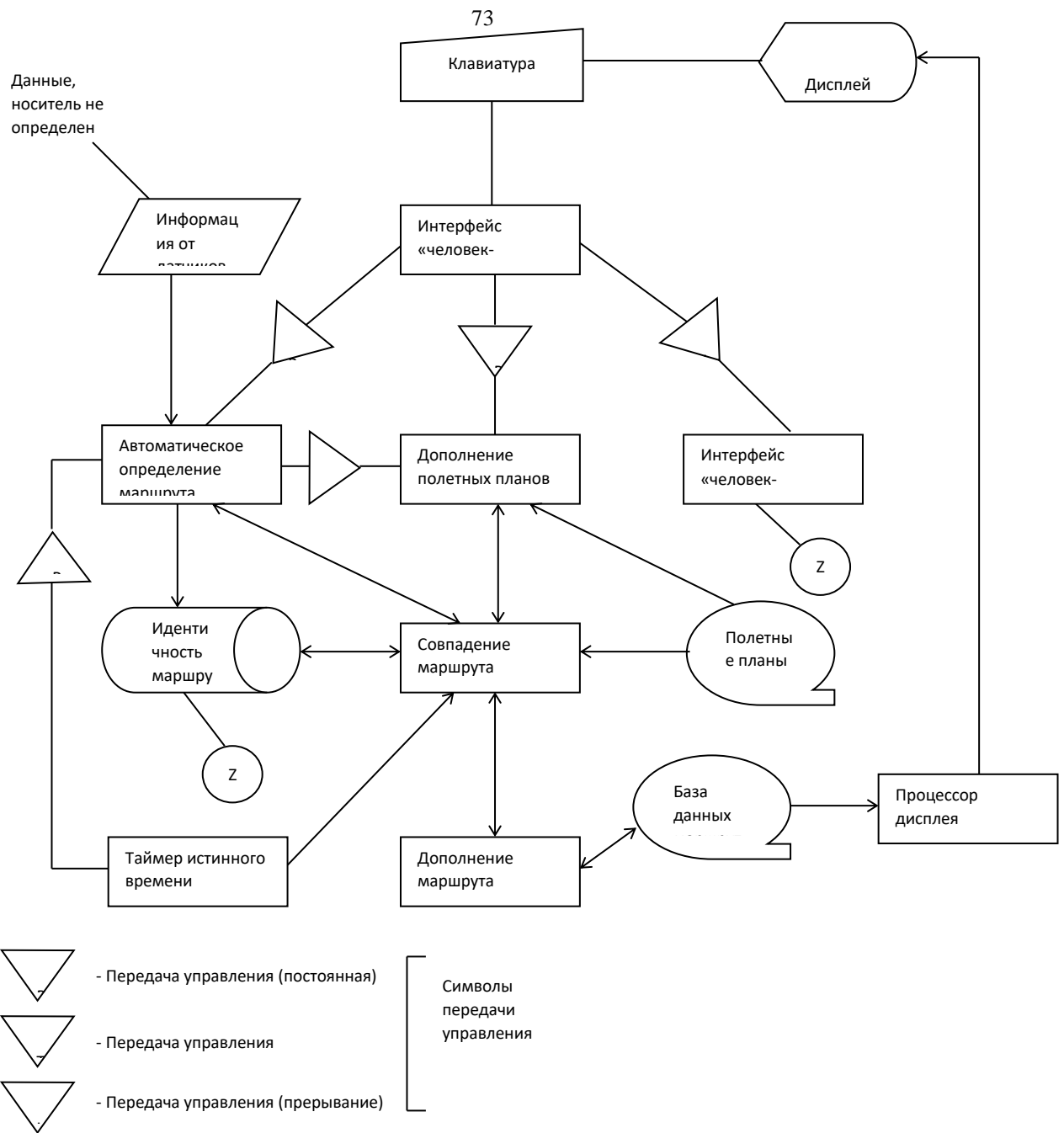
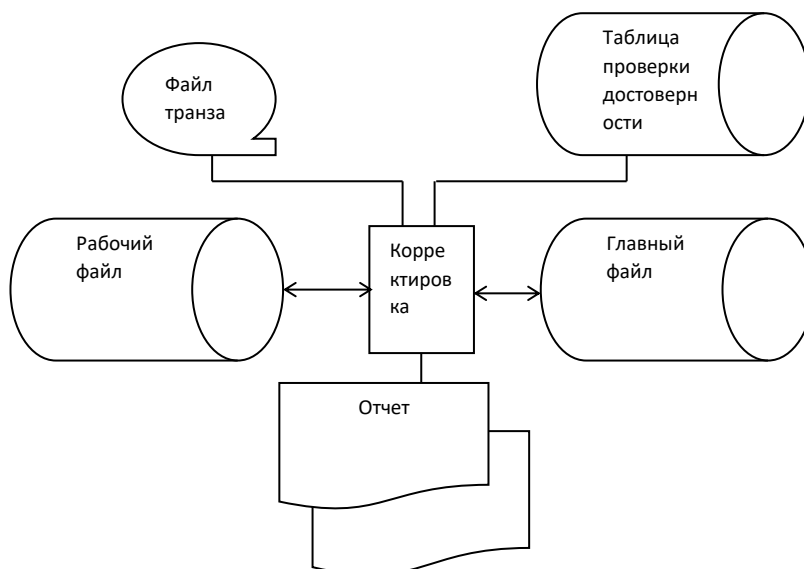


Рис. 9.4.10. Пример схемы взаимодействия программ



17. Постановка и модель решения задачи распределения ресурсов РП между НИР и ОКР.

8.1. Классификация рассматриваемых моделей

Объектом автоматизации, для которого будут рассмотрены математические модели для поиска оптимальных решений, является ОКБ, НИИ, опытный завод. Такое предприятие часто называют разрабатывающим предприятием (РП).

Конечной продукцией разрабатывающего предприятия является техническая документация, необходимая для изготовления образцов новой техники. Это изготовление может быть осуществлено либо на заводе, который входит в состав РП, либо на специализированных заводах, относящихся к заводам с серийным или массовым характером производства.

Одна из специфических особенностей РП – выполнение ими как ОКР, так и НИР. ОКР включаются в тематический план РП или директивно по указанию вышестоящей организации, или по предложению самого предприятия.

Источниками возникновения предложений на ОКР, включаемые в тематический план по предложению самого предприятия, являются другие РП отрасли, серийные заводы, организации-потребители. Предложения на их проведение могут поступать также от специалистов самого РП. В дальнейшем такие ОКР, в отличие от директивных, будем называть инициативными.

НИР выполняются для того, чтобы можно было создавать новые образцы техники, которые превосходят или не хуже лучших уже созданных аналогичных или подобных изделий.

Можно выделить следующие задачи управления РП, обеспечивающие его эффективно функционирование:

- задачи стратегического планирования, определяющие будущее развитие РП на период более пяти лет;
- задачи тактического планирования, определяющие работу РП на период от года до пяти лет;
- задачи оперативного управления РП в текущем году, квартале, месяце, как правило, ходом выполнения НИР и ОКР, включенным в годовой план работы РП.

Для оптимального решения этих задач в АСОИУ необходимо использовать комплекс математических моделей, которые связаны между собой информационно и функционально. Эта связь показана на рис. 8. 1..

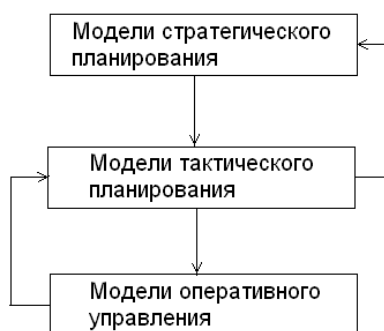


Рис. 1. Схема взаимосвязи моделей АСОИУ РП

Рис. 8.1.

Агрегированные модели распределения ресурсов РП между НИР и ОКР

Общая постановка задачи

ОКР включаются в тематический план РП или директивно по указанию вышестоящей организации, или по предложению самого предприятия.

Источниками возникновения предложений на ОКР, включаемые в тематический план по предложению самого предприятия, являются другие РП отрасли, серийные заводы, организации-потребители. Предложения на их проведение могут поступать также от специалистов самого РП. В дальнейшем такие ОКР, в отличие от директивных, будем называть инициативными.

Директивные и инициативные ОКР могут выполняться с превышением ранее достигнутого технического уровня и без этого превышения. В последнем случае содержанием ОКР является модификация ранее разработанных изделий. Такие ОКР требуют относительно небольших затрат трудовых и материальных ресурсов, но зато не способствуют научно-техническому прогрессу предприятия.

ОКР, имеющие целью создание изделий с улучшенными техническими характеристиками, обеспечивают научно-технический прогресс. Однако затраты на их проведение и вероятность достижения технического успеха в установленные сроки во многом зависят от уровня имеющегося на предприятии научно-технического задела, материализованного в виде макетных и экспериментальных образцов новых изделий, отчетов с предложениями и рекомендациями по использованию новых технических решений, элементной базы, материалов, технологических процессов, испытательного оборудования, стендов и т.п.

Научно-технический задел предприятия создается главным образом в процессе выполнения НИР.

Нежелателен как слишком низкий, так и слишком высокий уровень научно-технического задела и связанных с ним затрат ресурсов.

Во-первых это приводит к большим затратам на доработку опытных образцов изделий при попытке использовать недостаточно отработанные в процессе НИР

технические решения. По подсчетам экономистов устранение одной ошибки при выполнении ОКР требует в среднем в десять раз больше затрат, чем при выполнении НИР. С другой стороны, слишком большой уровень затрат на НИР для РП также не дает желаемого эффекта, что определяется главным образом следующим. Во-первых, на макетных и экспериментальных образцах практически невозможно провести полную отработку вновь разрабатываемых изделий, для этого необходимо выполнить конструкторскую проработку, изготовить и испытать опытные образцы этих изделий, т.е. провести ОКР.

Во-вторых, при большом объеме ресурсов, выделяемых на проведение НИР, будет создаваться такой научно-технический задел, что он не сможет полностью израсходоваться в ОКР из-за недостатка необходимых для этого ресурсов. В результате предприятие будет работать, как говорят, “на полку”, т.е. без реализации НИР в конкретных ОКР.

Таким образом, при управлении разработками на РП возникает задача определения наиболее рациональной пропорции между затратами ресурсов на НИР и ОКР, при которой, с одной стороны, будут обеспечены необходимые темпы научно-технического прогресса РП, с другой – не будет непроизводительного расходования ресурсов на НИР, что позволит в конечном итоге ускорить проведение ОКР.

Эффект от проведения НИР обычно сказывается не непосредственно в планируемом периоде, а спустя некоторый промежуток времени, определяемый конкретными особенностями процесса выполнения НИР в ОКР на предприятии. Поэтому решение поставленной задачи требует рассмотрения не одного, а нескольких плановых периодов.

Значительная часть РП обладает следующими особенностями:

- большое число одновременно проводимых НИР и ОКР;
- конструктивно-технологическая однородность разрабатываемых изделий;
- использование результатов НИР при выполнении нескольких ОКР;
- относительно небольшая продолжительность отдельных НИР (2-3 года) и ОКР (до 5 лет);
- выполнение НИР и ОКР в одних и тех же конструкторских и экспериментальных подразделениях предприятия.

Эти особенности позволяют, во-первых, рассматривать данную задачу на агрегированном уровне в целом по РП без учета специфических особенностей конкретных разработок, во-вторых, учитывать только один вид ресурсов, а не вектор ресурсов, в-третьих, что очень важно, они позволяют ограничиться при построении моделей только ресурсным аспектом, считая, что непосредственное увеличение технического уровня разрабатываемых при выполнении конкретных ОКР изделий авиационной техники происходит постоянно с некоторым темпом, определяемым требованиями заказчиков, особенностями РП, качеством отбора и

выполнения НИР и ОКР. Последнее обстоятельство дает возможность в качестве критерия оптимальности при решении данной задачи взять среднее количество ОКР, которые выполняются в течение рассматриваемого планового интервала времени.

Рассмотрим две модели для решения поставленной задачи.

Модель на основе введенной меры научно-технического задела

Пусть для каждого интервала времени t планового периода T на основании статистического анализа результатов предшествующей деятельности предприятия и прогнозных исследований определены количества директивных ОКР D_t , а также максимально возможные количества выполняемых инициативных ОКР U_t^0 и НИР H_t^0 ($t = \overline{1, T}$).

Известно количество ресурсов, идущих на проведение одной директивной ОКР, имеющей задел r_D' и не имеющей задела r_D'' .

Известны также соответствующие значения ресурсов r_u' и r_u'' , необходимых для выполнения инициативных ОКР. Очевидно, что $r_D' < r_D''$ и $r_u' < r_u''$, т.е. научно-технический задел снижает количество ресурсов, требуемых для выполнения ОКР.

Пусть известно также количество ресурсов r_H , необходимое для выполнения одной НИР.

Обозначим через D_t' (U_t') количество директивных (инициативных) ОКР, имеющих задел, а через D_t'' (U_t'') - соответственно, не имеющих задела.

Обозначим через R_t количество ресурсов, которыми располагает предприятие в период t ($t = \overline{1, T}$).

При принятых обозначениях задачу оптимизации структуры затрат ресурсов РП можно сформулировать следующим образом.

Требуется при заданных ограничениях на ресурсы РП R_t ($t = \overline{1, T}$) выбрать в каждом плановом периоде t такое количество НИР H_t , при котором за время T было бы выполнено максимально возможное количество ОКР.

При этом уровень научно—технического задела на начало следующего $(T + 1)$ -го планового периода должен быть бы не ниже некоторого заданного уровня.

Математическая постановка задачи имеет следующий вид:

$$W = \max_{H_1, H_2, \dots, H_T} \sum_{t=1}^T (D_t + U_t) \quad (1)$$

при следующих ограничениях:

$$\begin{aligned} D_t' r_D' + D_t'' r_D'' + U_t' r_u' + U_t'' r_u'' + r_H H_t &\leq R_t; \\ U_t &= U_t' + U_t'' \leq U_t^0; D_t' + D_t'' = D_t; H_t \leq H_t^0; \end{aligned} \quad (2)$$

$$Z_1 = A; Z_{T+1} \geq B; t = \overline{1, T},$$

где Z_1 и Z_{T+1} - уровни научно-технического задела, имеющиеся на предприятии, на начало и на конец планового горизонта соответственно.

В качестве количественной меры научно—технического задела, имеющегося на предприятии по данному техническому направлению к началу t -го планового периода, используем величину P_t ($0 \leq P_t \leq 1$), представляющую собой вероятность того, что случайно выбранная ОКР из числа директивных D_t или инициативных U_t имеет научно—технический задел.

При большом количестве одновременно выполняемых ОКР эту величину можно интерпретировать как долю директивных или инициативных ОКР t -го планового периода, для

которых ранее были выполнены соответствующие НИР.

Отражая свойства научно-технического задела, величина P_t зависит от:

- уровня задела к началу $(t-1)$ -го периода;
- является возрастающей функцией количества ресурсов, выделенных в $(t-1)$ -м плановом периоде на НИР;
- является убывающей функцией времени ввиду морального старения неиспользованного задела;
- является убывающей функцией количества выполненных ОКР в $(t-1)$ -м периоде с заделом.

Представим величину P_t следующим образом:

$$P_t = \begin{cases} 0, & \text{если } \varphi_t < 0, \\ \varphi_t, & \text{если } 0 \leq \varphi_t \leq 1, \\ 1, & \text{если } \varphi_t > 1, \end{cases} \quad (3)$$

где $\varphi_t = P_{t-1} - \alpha - \beta(U'_{t-1} + D'_{t-1}) + \gamma H_{t-1}$,

α - величина, характеризующая старение научно-технического задела за один плановый период;

β - коэффициент, отражающий эффект расходования научно-технического задела при выполнении одной ОКР;

γ - коэффициент, отражающий средний вклад одной НИР в создание научно-технического задела.

Пусть на начале t -го планового периода создан задел, характеризующийся величиной P_t , и, значит, для выполнения директивных ОКР в этом периоде потребуется среднее количество ресурса, равное $D_t(P_t r'_D + (1 - P_t)r''_D)$.

Будем считать, что директивные ОКР всегда можно выполнить даже при полном отсутствии научно-технического задела. Тогда, возможно, еще останется неиспользованной часть ресурса R_t , которую можно направить или на инициативные ОКР, или на НИР, или на те и другие вместе. Пусть в период t выполняются НИР в количестве H_t ($0 \leq H_t \leq H_t^0$) и, значит, остается неиспользованным количество ресурса ψ_t :

$$\psi_t = R_t - D_t(P_t r'_D + (1 - P_t)r''_D) - r_H H_t. \quad (4)$$

Это количество ресурсов можно израсходовать на инициативные ОКР, среднее количество которых в период t будет равно:

$$U'_t = \begin{cases} P_t U_t^0, & \text{если } \psi_t > P_t U_t^0 r'_U; \\ \frac{\psi_t}{r'_U}, & \text{если } \psi_t \leq P_t U_t^0 r'_U; \end{cases} \quad (5')$$

$$U''_t = \begin{cases} 0, & \text{если } \psi_t \leq P_t U_t^0 r'_U; \\ \frac{\psi_t - P_t U_t^0 r'_U}{r''_U}, & \text{если } \psi_t > P_t U_t^0 r'_U. \end{cases} \quad (5'')$$

На конец t -го периода, или, что то же самое, на начало $t+1$ -го периода будет создан при этом научно-технический задел, измеряемый величиной P_{t+1} , определяемый следующей формулой:

$$P_{t+1} = P_t - \alpha - \beta(U'_t + P_t D_t) + \gamma H_t. \quad (6)$$

Так как по сделанному выше предположению директивные темы всегда выполняются, то, с учетом формул (3) - (6) задачу (1) - (2) можно сформулировать следующим образом:

$$W = \max_{H_1, H_2, \dots, H_T} \sum (U'_t + U''_t) \quad (7)$$

при условиях (4), (5), (6) и

$$U'_t + U''_t \leq U^0_t; H_t \leq H^0_t; P_1 = P^{HAq}; P_{T+1} \geq P^{KOH}. \quad (8)$$

Для решения задачи (7) при условиях (4), (5), (6), (8) выведем необходимое рекуррентное соотношение.

Перенумеруем интервалы времени в обратном порядке, т.е. конечный интервал T получит номер 1, а начальный - номер T .

Обозначим через $f_j(P_j)$ максимально возможное (оптимальное) количество директивных и инициативных ОКР, включенных в план с j -го до 1-го (в новой нумерации) периода при условии, что на начало j -го периода уровень научно-технического задела равен P_j .

Для $j=1$ функция $f_1(P_1)$ имеет вид:

$$f_1(P_1) = \max_{0 \leq H_1 \leq H^0_1} \{D_1 + U'_1(P_1, H_1) + U''_1(P_1, H_1)\}, \quad (9)$$

где $U'_1(P_1, H_1)$ и $U''_1(P_1, H_1)$ определяются формулами (5') и (5'').

Так как уровень задела на конец 1-го периода известен и равен $P_0 = P^{KOH}$, то при построении функции $f_1(P_1)$, варьируя H_1 , надо проверять выполнимость неравенства

$$P_1 - \alpha - \beta(U'_1(P_1, H_1) + PD_1) + \gamma H_1 \geq P^{KOH}. \quad (10)$$

Это неравенство ограничивает снизу величину H_1 .

Пусть функция $f_1(P_1)$ построена. Найдём выражение для величины $f_2(P_2)$, т.е. максимальное количество ОКР, которое можно выполнить за два периода, считая от конца планового горизонта, при уровне задела на начало 2-го периода, равном P_2 :

$$f_2(P_2) = \max_{\substack{0 \leq H_1 \leq H^0_1 \\ 0 \leq H_2 \leq H^0_2}} \{D_2 + U'_2(P_2, H_2) + U''_2(P_2, H_2) + D_1 + U'_1(P_1, H_1) + U''_1(P_1, H_1)\}, \quad (11)$$

причем между величинами P_2 и P_1 существует следующая связь:

$$P_1(P_2, H_2) = P_2 - \alpha - \beta(U'_2(P_2, H_2) + P_2 D_2) + \gamma H_2. \quad (12)$$

В выражении (11) с ростом H_2 первая группа слагаемых уменьшается, а вторая - увеличивается. С уменьшением H_2 происходит обратное явление.

Нетрудно заметить, что выражение (11) можно переписать следующим образом:

$$f_2(P_2) = \max_{0 \leq H_2 \leq H^0_2} \{D_2 + U'_2(P_2, H_2) + U''_2(P_2, H_2) + f_1(P_1(P_2, H_2))\}, \quad (13)$$

где функция $P_1(P_2, H_2)$ определяется выражением (12).

Рассуждая аналогично, можно построить следующие рекуррентные формулы для решения поставленной задачи:

$$f_j(P_j) = \max_{0 \leq H_j \leq H^0_j} \{D_j + U'_j(P_j, H_j) + U''_j(P_j, H_j) + f_{j-1}(P_{j-1}(P_j, H_j))\}, \quad (14)$$

$$P_{j-1}(P_j, H_j) = P_j - \alpha - \beta(U'_j(P_j, H_j) + P_j D_j) + \gamma H_j. \quad (15)$$

Пусть для всех $j = 1, 2, \dots, T$ построены функции $f_j(P_j)$, причем наряду с построением этих функций строятся и функции $H_j^*(P_j)$, т.е. фиксируются те количества НИР в каждом плановом периоде, при которых достигается максимум выражения (14).

Оптимальные пропорции между затратами ресурсов на НИР и ОКР могут быть теперь определены следующим образом.

По известной величине P^{HAC} — уровню задела на начало рассматриваемого планового горизонта в таблице функций H_t^* ($t = \overline{1, T}$) находим оптимальное значение числа НИР, выполняемых в первом (возвращаясь опять к исходной нумерации периодов) периоде H_1 .

Умножив эту величину на r_H , находим средние затраты на НИР в этом периоде. Остальные $(R_1 - H_1 r_1)$ ресурсы идут на ОКР, причем $D_1(P_1 r_D' + (1 - P_1) r_D'')$ ресурса расходуется на директивные ОКР, а остаток - на инициативные ОКР. Для остальных плановых периодов эти расчеты носят оценочный характер, так как после окончания первого периода все расчеты необходимо повторить с учетом дополнительной информации о фактических затратах на НИР и ОКР с заделом и без него, о числе директивных и инициативных ОКР.

8.2.3. Модель на основе временной зависимости затрат ресурсов на НИР и ОКР

Пусть в период t в подразделении выполняются НИР в объеме $H_t r_t^H$, где H_t - число выполняемых НИР, r_t^H - средние затраты на одну НИР. Тогда на ОКР в этом же периоде остается $R_t - H_t r_t^H$ ресурсов, где R_t - ресурсы подразделения в период t . Если r_t^0 - средние затраты на одну ОКР в единицу времени, то в период t в подразделении могут выполняться работы в среднем по $\frac{R_t - H_t r_t^H}{r_t^0}$ ОКР.

Если в одном и том же подразделении предприятия выполняются и НИР и ОКР, то, очевидно, что чем больше было затрачено ресурсов на отработку макетных и экспериментальных образцов в процессе НИР, тем меньше потребуется ресурсов на отработку опытных образцов при выполнении ОКР. Это влияние НИР на ОКР может проявляться, вообще говоря, не сразу, а спустя некоторый период, определяемый продолжительностью выполнения соответствующих этапов ОКР в других подразделениях - соисполнителях этих ОКР.

Таким образом, величина r_t^0 есть некоторая функция затрат ресурсов, выделенных на НИР к периодам назад, т.е.

$$r_t^0 = f_{t-k-1}(H_{t-k-1} r_{t-k-1}^H). \quad (1)$$

Исходя из характера принимаемого во внимание влияния НИР на ОКР, можно предположить, что эта функция представляет собой перевернутую S-образную кривую. С достаточной для практики точностью можно аппроксимировать эту функцию перевернутой логистической кривой (рис. 8.2).

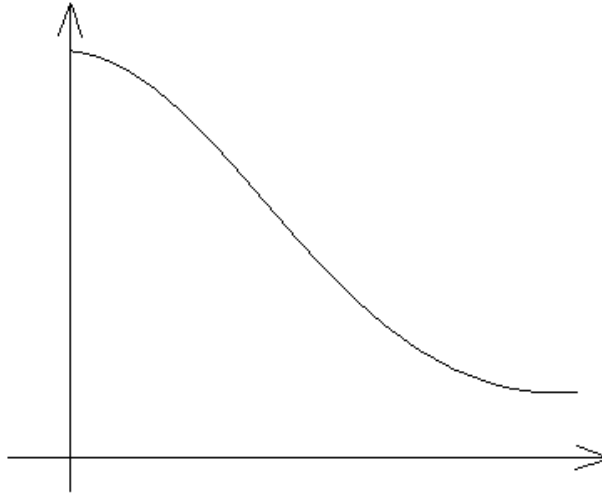


Рис. 1. Вид логистической функции

Рис. 8.2.

Считая, что число выполняемых в каждом периоде НИР известно (оно обычно мало меняется от периода к периоду), задачу распределения ресурсов между НИР и ОКР можно теперь сформулировать следующим образом:

$$W = \max_{H_1 r_1^H, \dots, H_T r_T^H} \sum_{t=1}^T \frac{R_t - H_t r_t^H}{f_{t-k-1}(H_{t-k-1} r_{t-k-1}^H)}; \quad (2)$$

$$0 < H_t r_t^H \leq R_t, t = \overline{1, T} \quad (3)$$

$$r_1^0 = f_{-k-1}(H_{-k-R-1} r_{k-1}^H) = B_1; \dots; r_{k+1}^0 = f_{-k}(H_{-1} r_{-1}^H) = B_{k+1}; \quad (4)$$

$$r_{T+1}^0 = \xi_1; \dots; r_{T+k+1}^0 = \xi_{k+1}. \quad (5)$$

Максимизируемый функционал (2) представляет собой среднее количество ОКР, выполненных за T периодов, где T - рассматриваемый плановый горизонт. В качестве управляемых переменных выступают затраты на НИР $H_t r_t^H$ в периоды $t=1, 2, \dots, T$. Условие (3) означает, что в любой период t суммарные затраты на НИР не могут превзойти ресурсные возможности подразделения. Условие (4) означает, что известны затраты ресурсов на НИР в периоды, отстоящие от исходного ($t=1$) на $1, 2, \dots, (k+1)$ интервалов времени назад. Наконец, условие (5) означает, что известны средние затраты ресурсов на одну ОКР в периоды $T+1, T+2, T+k+1$. Знание их необходимо для того, чтобы ограничить нижний предел величины $H_t r_t^H$ для $t=T-k, \dots, T$. Из (2) непосредственно следует, что при отсутствии условий (5) значения $H_t r_t^H$ для $t=T-k, \dots, T$, оптимизирующие (2), равны 0, т.е. в эти периоды не будут совсем проводиться НИР. Для заданного планового горизонта можно произвольно выбирать значения $H_t r_t^H$ только для первых $(T-k-1)$ периодов, так как значения $H_t r_t^H$ для остальных $(k+1)$ периодов определены условиями (5). Эти $H_t r_t^H$ изменяют значения средних затрат на одну ОКР для последних $(T-k-1)$ периодов, так как значения средних затрат на одну ОКР для первых $(k+1)$ периодов определены условиями (4). Поэтому задача (2)-(5) имеет нетривиальное решение только для $T > k+1$.

При построении рекуррентного соотношения используем метод математической индукции. Для этого обозначим через $\psi_t(r_{t+1}^0, \dots, r_{t+k+1}^0)$ максимально возможное среднее количество ОКР, которое можно выполнить за t периодов при условии, что в $(t+1), (t+2), \dots, (t+k+1)$ периодах будет создан научно-технический задел, обеспечивающий возможность выполнения ОКР в эти периоды со средними затратами ресурсов, соответственно равными $r_{t+1}^0, \dots, r_{t+k+1}^0$.

Очевидно, что для $t=k+1$ функция $\psi_t(r_{t+1}^0, \dots, r_{t+k+1}^0)$ определяется следующим образом:

$$\psi_{k+1}(r_{k+2}^0, \dots, r_{2k+2}^0) = \frac{R_1 - \varphi_1(r_{k+2}^0)}{B_1} + \frac{R_2 - \varphi_2(r_{k+3}^0)}{B_2} + \dots + \frac{R_{k+1} - \varphi_{k+1}(r_{2k+2}^0)}{B_{k+1}}, \quad (6)$$

где $\varphi_{k+1}(\dots)$ - функция, обратная (1).

Для $t=k+2$ эта функция имеет вид

$$\psi_{k+2}(r_{k+3}^0, \dots, r_{2k+3}^0) = \max_{0 \leq H_1 r_1^H \leq R_1} \left\{ \frac{R_1 - H_1 r_1^H}{B_1} + \frac{R_2 - \varphi_2(r_{k+3}^0)}{B_2} + \dots + \frac{R_{k+1} - \varphi_{k+1}(r_{2k+2}^0)}{B_{k+1}} + \frac{R_{k+2} - \varphi_{k+2}(r_{2k+3}^0)}{f_1(H_1 r_1^H)} \right\}. \quad (7)$$

Сравнивая выражения (6) и (7), замечаем, что в них второе, третье и т.д. до $(k+1)$ -го слагаемые совпадают. Если в (6) положить $r_{k+2}^0 = f_1(H_1 r_1^H)$, то и первые слагаемые в (6) и (7) совпадут, так как $\varphi_1(r_{k+2}^0 = f_1(H_1 r_1^H)) = H_1$.

Отсюда следует, что

$$\psi_{k+2}(r_{k+3}^0, \dots, r_{2k+3}^0) = \max_{0 \leq H_1 r_1^H \leq R_1} \left\{ \psi_{k+1}(r_{k+2} = f_1(H_1 r_1^H), r_{k+3}^0, \dots, r_{2k+2}^0) + \frac{R_{k+2} - \varphi_{k+2}(r_{2k+3}^0)}{f_1(H_1 r_1^H)} \right\}. \quad (8)$$

Рассуждая аналогично, по методу математической индукции можно доказать, что для любого t ($k+1 < t \leq T$) справедливо следующее рекуррентное соотношение:

$$\psi_t(r_{t+1}^0, \dots, r_{t+k+1}^0) = \max_{0 \leq H_{t-k-1} r_{t-k-1}^H \leq R_{t-k-1}} \left\{ \psi_{t-1}(r_t^0 = f_{t-k-1}(H_{t-k-1} r_{t-k-1}^H), r_{t+1}^0, \dots, r_{t+k}^0) + \frac{R_t - \varphi_t(r_{t+k+1}^0)}{f_{t-k-1}(H_{t-k-1} r_{t-k-1}^H)} \right\} \quad (9)$$

Задачу (2) - (5) можно решить теперь следующим образом. Используя формулу (6), вычисляем функции $\psi_{k+1}(r_{k+2}^0, \dots, r_{2k+2}^0)$ для всех значений $r_{\min}^0 \leq r_{k+1+j}^0 \leq r_{\max}^0$, $j = 1, \dots, k+1$, определенных с заданной дискретностью. После этого, используя формулу (8), а затем последовательно формулу (9), вычисляем значения функций $\psi_t(r_{t+1}^0, \dots, r_{t+k+1}^0)$, полагая $t=k+2, \dots, T-1, T$. Для каждого набора переменных $(r_{t+1}^0, \dots, r_{t+k+1}^0)$ фиксируем значение $\hat{H}_{t-k-1} r_{t-k-1}^H$, при котором достигается максимум в выражении (9). Полагая в функции $\psi_T(r_{T+1}^0, \dots, r_{T+k+1}^0)$ значения переменных равными соответственно ξ_1, \dots, ξ_{k+1} , находим решение задачи (2) - (5) и искомое значение $\tilde{H}_{T-k-1} r_{T-k-1}^H$, при котором достигается максимум выражения (9). После этого вычисляем значение $\tilde{r}_T = f_{T-k-1}^0(\tilde{H}_{T-k-1} r_{T-k-1}^H)$. По набору переменных $(r_T, \xi_1, \dots, \xi_k)$ в таблице, определяющей функцию $\psi_{T-1}(r_T^0, \dots, r_{T-k}^0)$, находим соответствующее значение $\tilde{H}_{T-k-2} r_{T-k-2}^H$. Затем вычисляем $\tilde{r}_{T-1} = f_{T-k-2}^0(\tilde{H}_{T-k-2} r_{T-k-2}^H)$ и в таблице, определяющей функцию $\psi_{T-2}(r_{T-1}^0, \dots, r_{T-k-1}^0)$, находим соответствующее значение $\tilde{H}_{T-k-2} r_{T-k-2}^H$, и т.д. до тех пор, пока не дойдем до $\psi_{k+2}(\dots)$ и не получим значение $H_1 r_1^H$.

В результате описанной процедуры получаем искомые значения $\tilde{H}_1 r_1^H, \dots, \tilde{H}_{T-k-1} r_{T-k-1}^H$ для первых $(T-k-1)$ периодов. Остальные значения $\tilde{H}_{T-k-1-j}, j = \overline{1, k+1}$ вычисляются путем решения уравнений $\tilde{H}_{T-k-1+j} r_{T-k-1+j}^H = \varphi_j(\xi_j)$, $j = \overline{1, k+1}$, где ξ_j определяются условием (5).

18. Модели этапа исходного планирования разработок: модели оценки технического уровня и распределения ресурсов по этапам выполнения предстоящей разработки

Модели этапа исходного планирования разработок

8.3.1. Модель оценки технического уровня предстоящей разработки

Технический уровень изделия – это относительная характеристика качества этого изделия, основанная на сопоставлении значений показателей (параметров), характеризующих техническое и эргономическое совершенство оцениваемого изделия с соответствующими базовыми (эталонными) изделиями.

В процессе определения критерия оценки технического уровня учитывают влияние не одного какого-либо параметра, а совокупность нескольких основных, важнейших параметров, так как улучшение одного из них может привести к ухудшению другого, что в итоге может привести к изменению значения выбранного критерия.

Перечень таких параметров, как правило, устанавливает разработчик изделия, возможно совместно с другими заинтересованными организациями. При этом, очевидно, каждый параметр следует выбирать таким образом, чтобы с его увеличением значение критерия увеличивалось бы.

Всю совокупность выбранных параметров рассматриваемого изделия представим в виде вектора

$$Y_u = (y_{1u}, y_{2u}, \dots, y_{nu})^T, \quad (1)$$

где y_{iu} - значения параметра i , ($i = \overline{1, n}$) для изделия u .

Составим следующую матрицу значений параметров группы из m аналогичных изделий, которая характеризует достигнутый отечественный или мировой уровень аналогичных изделий:

$$Y = \begin{pmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & \dots & y_{nm} \end{pmatrix}. \quad (2)$$

Переходя к лучшим значениям параметров в выражении (2), получим вектор

$$Y_0 = (y_1^0, y_2^0, \dots, y_n^0)^T, \quad (3)$$

где $y_i^0 = \max_{j=1, m} \{y_{ij}\}$, $i = \overline{1, n}$.

Вектор Y_0 характеризует технический уровень некоторого идеального, эталонного изделия.

Обобщенный показатель технического уровня исследуемого изделия K_{TY}^u можно теперь определить следующим образом:

$$K_{TY}^u = \sum_{i=1}^n a_i \frac{y_{iu} - y_i^0}{\Delta_i}, \quad (4)$$

где:

Δ_i - предельное отклонение i -го параметра, т.е. разность между наибольшим и наименьшим значениями i -го параметра, взятыми по всей совокупности параметров изделий, включая и рассматриваемое изделие. Если $y_{iu} - y_i^0 = 0$, $\Delta_i = 0$, то $\frac{y_{iu} - y_i^0}{\Delta_i} = 0$;

a_i - весовой коэффициент, отражающий неравнозначность вклада отдельных параметров в оценку технического уровня изделия. Полагают, что $\sum_{i=1}^n a_i = 1$.

Если подсчитанный по формуле (4) показатель $K_{TY}^u > 0$, то это означает, что технический уровень оцениваемого изделия выше технического уровня аналогичных изделий взятых для сравнения. Если же $K_{TY}^u < 0$, то это означает, что технический уровень оцениваемого изделия ниже технического уровня аналогов.

В формуле (4) неизвестными являются численные значения весовых коэффициентов a_i , $i = \overline{1, n}$.

Для их определения может быть применен метод парных сравнений. Его сущность состоит в том, что каждому эксперту, выбранному для определения важности отдельных показателей в итоговой оценке технического уровня изделия, предлагается сравнивать относительную важность пар различных параметров. Для этого составляют квадратные матрицы парных сравнений, в которых все параметры записываются в одном и том же порядке дважды: в верхней строке и в крайнем левом столбце каждой матрицы. В клетках на пересечении i – й строки и j – го столбца ($i, j = \overline{1, n}$) каждый эксперт проставляет числа x_{ij} по следующим правилам:

$$x_{ij} = \begin{cases} 1, & \text{если параметр } i \text{ , более предпочтителен, чем параметр } j; \\ 0,5, & \text{если оба параметра } i \text{ и } j \text{ равноценны;} \\ 0, & \text{если параметр } j \text{ более предпочтителен, чем параметр } i. \end{cases}$$

Каждая пара параметров может сравниваться дважды или только один раз. Последний вариант встречается на практике, поэтому его и рассмотрим. Он предполагает, что справедливо следующее очевидное равенство: $x_{ij} + x_{ji} = 1$.

Полученную от эксперта с номером l , $l = \overline{1, L}$ матрицу парных сравнений обозначим $X_l = \|x_{ij}^l\|$.

Значения весовых коэффициентов a_i^l , $i = \overline{1, n}$ параметров в формуле (4) для эксперта с номером l найдем на основе обработки полученной матрицы $X_l = \|x_{ij}^l\|$, воспользовавшись решением так называемой задачи о лидере.

Обозначим $A_t^l = (a_{1t}^l, a_{2t}^l, \dots, a_{nt}^l)$ вектор весовых коэффициентов параметров для эксперта с номером l , полученный на шаге вычислений $t = 1, 2, \dots$

Для вычисления компонент этого вектора воспользуемся следующей формулой:

$$A_t^l = \frac{1}{\lambda_t^l} X_l \cdot A_{t-1}^l, \quad (5)$$

где величина λ_t^l определяется по следующей формуле:

$$\lambda_t^l = \sum_{i=1}^n \sum_{j=1}^n x_{ij}^l \cdot a_{i(t-1)}^l. \quad (6)$$

На первом шаге вычислений полагают все элементы вектора A_0^l равными единице.

По определению, матрица $X_l = \|x_{ij}^l\|$ неотрицательная. Поэтому согласно теореме Перрона – Фробениуса при стремлении числа шагов вычислений t к бесконечности, величина λ_t^l стремится к собственному числу λ_0^l матрицы X_l , а вектор весовых коэффициентов $A_t^l = (a_{1t}^l, a_{2t}^l, \dots, a_{nt}^l)$ стремится к собственному вектору этой матрицы, соответствующему максимальному собственному числу λ_0^l , т.е. можно записать:

$$A^l = \lim_{t \rightarrow \infty} A_t^l, \quad \sum_{i=1}^n a_i^l = 1, \quad (7)$$

где $A^l = (a_1^l, a_2^l, \dots, a_n^l)$.

Таким образом, последовательно применяя формулы (5) и (6), находим L векторов A^1, A^2, \dots, A^L , компонентами которых являются весовые коэффициенты параметров в формуле (4).

Необходимо теперь установить степень согласованности мнений экспертов относительно полученных значений компонент векторов A^1, A^2, \dots, A^L .

Одним из возможных подходов для определения степени согласованности мнений экспертов является подход, основанный на использовании метода ранговой корреляции и вычислении так называемого коэффициента конкордации W .

Для его вычисления перейдем от значений коэффициентов относительной важности $(a_1^l, a_2^l, \dots, a_n^l)$ к рангам этих критериев r_i^l . При этом ранг, равный единице, присваивается параметру, у которого соответствующий ему весовой коэффициент наибольший. Ранги $1, 2, \dots, n$ присваиваются параметрам по мере уменьшения их весовых коэффициентов. Ранг n присваивается параметру, имеющему наименьший весовой коэффициент.

Коэффициент конкордации W рассчитывается по следующей формуле, которую предложил Кендалл:

$$W = \frac{12 \cdot S}{L^2 (n^3 - n)}, \quad (8)$$

где

$$S = \sum_{i=1}^n \left(\sum_{l=1}^L r_i^l - A \right)^2, \quad (9)$$

$$A = \frac{1}{n} \sum_{i=1}^n \sum_{l=1}^L r_i^l; \quad (10)$$

L – количество экспертов,

n – количество локальных критериев,

r_i^l – ранг локального критерия с номером i ($i = 1, 2, \dots, n$) согласно мнению эксперта с номером l ($l = 1, 2, \dots, L$)

Коэффициент конкордации W меняется от 0 до 1, причем, если он равен 1, то оценки всех экспертов полностью совпадают; если же он равен 0, то связь между оценками отсутствует.

Если $W \leq 0.2$, то согласованность оценивается как низкая; если $0.2 \leq W \leq 0.3$ – удовлетворительная и, если $W > 0.3$ – высокая.

Низкий коэффициент согласованности указывает либо на неудачный выбор ранжируемых факторов, либо на то, что мнения экспертов по данному вопросу резко расходятся. В этом случае необходимо провести новый экспертный опрос, возможно изменив состав экспертов.

Если гипотеза о согласии экспертов принимается, то вычисляются обобщенные итоговые значения весовых коэффициентов относительной важности параметров.

Вначале находим вектор $\bar{A} = (\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n)$ усредненных по всем экспертам значений весовых коэффициентов параметров в формуле (4).

Величины \bar{a}_i , $i = 1, n$ можно вычислить по следующим формулам:

$$\bar{a}_i = \frac{\sum_{l=1}^L a_i^l}{L}, \quad i = 1, n. \quad (11)$$

Формулы (11) предполагает, что все эксперты по своему уровню равноценны, т.е. их мнения равнозначны. Если это не так, т.е. среди экспертов есть специалисты различной квалификации, то вместо формул (11) более целесообразно применить формулы

$$\bar{a}_i = \frac{\sum_{l=1}^L \beta_l a_i^l}{L}, \quad i = 1, n, \quad (12)$$

где β_l – коэффициент, отражающий степень подготовленности эксперта с номером l , ($l = 1, L$),

причем $\sum_{l=1}^L \beta_l = 1$. Для определения значений коэффициентов β_l , ($l = 1, L$) можно применить

методику, изложенную выше.

Полученные по формулам (11) или (12) весовые коэффициенты не отвечают условию нормировки, так как в общем случае $\sum_{i=1}^n \bar{a}_i \neq 1$. Поэтому для получения условия нормировки следует перейти от вектора \bar{A} к требуемому вектору параметров $A = (a_1, a_2, \dots, a_n)$, вычислив значения компонент этого вектора по формулам:

$$a_i = \frac{\bar{a}_i}{\sum_{i=1}^n \bar{a}_i}, \quad i = \overline{1, n}. \quad (13)$$

Модель распределения ресурсов по этапам выполнения
Разработки

Пусть в результате предварительного исследования технической сущности предстоящей разработки определены основные этапы ее выполнения, число которых обозначим N .

Нормативную продолжительность выполнения каждого n -го этапа ($n = \overline{1, N}$) обозначим a_n .

За счет привлечения дополнительных ресурсов продолжительность этапа a_n может быть уменьшена на величину не более чем A_n , причем, очевидно, $0 \leq A_n < a_n$.

Если нормативная продолжительность этапа a_n уменьшена на величину z_n ($0 < z_n \leq A_n$), то его стоимость увеличивается на величину $C_n(z_n)$, где $C_n(z_n)$ - известная возрастающая функция аргумента. Таким образом, стоимость выполнения n -го этапа ($n = \overline{1, N}$) равна $C_n^0 + C_n(z_n)$, где C_n^0 - стоимость этапа при его нормативной продолжительности a_n .

Превышение заданного срока выполнения разработки на величину x может привести к экономическим санкциям в виде штрафа в размере $P(x)$.

Одновременно, досрочное выполнение разработки может быть связано с сокращением продолжительности выполнения отдельных этапов и, значит, их удорожанием, а, кроме того, может привести к экономическим потерям из-за необходимости хранения готовой продукции на складе, отвлечения разработчиков от выполнения других разработок и т.п. Эти потери обозначим $K(x)$, где x - время «пролеживания» результатов выполнения разработки.

Исходя из сказанного, ожидаемая стоимость выполнения разработки R может быть определена следующим образом:

$$R(t, z_1, z_2, \dots, z_N) = \sum_{n=1}^N (C_n^0 + C_n(z_n)) + \psi(t, z_1, z_2, \dots, z_N), \quad (1)$$

$$\text{где } \psi(t, z_1, z_2, \dots, z_N) = \begin{cases} K(x), & x > 0; \\ 0, & x = 0; \\ P(x), & x < 0, \end{cases}$$

$x = t - \sum_{n=1}^N (a_n - z_n)$ - величина запаздывания или опережения заданного срока выполнения

разработки;

t - интервал времени между началом выполнения разработки и требуемым (директивным) сроком ее завершения.

Задача состоит в том, чтобы при заданном t найти такие значения величин z_1, z_2, \dots, z_N ,

которые минимизировали бы ожидаемую стоимость выполнения разработки.

Решение поставленной задачи можно найти, построив динамическую оптимизационную модель.

Занумеруем все этапы разработки в обратном порядке, так что номер N получит первый этап, номер $(N - 1)$ - второй этап и т.д.

Пусть после завершения $(n + 1)$ -го этапа осталось t единиц времени до установленного срока окончания разработки. Тогда величина $t - (a_1 + a_2 + \dots + a_n)$ - это время, которое можно потратить на устранение различного рода задержек в выполнении оставшихся этапов. Обозначим эту величину x_n .

Перед началом выполнения n -го этапа величина x_n может быть увеличена на $0 < z_n \leq A_n$, а в процессе выполнения за счет различных случайных факторов уменьшена в среднем на величину d_n . Поэтому соотношение между величинами x_n и x_{n-1} может быть записано в следующем виде:

$$x_{n-1} = x_n + z_n - d_n, \quad n = \overline{1, N}. \quad (2)$$

Очевидно, что при этих обозначениях величина x_0 равна неиспользованному резерву времени, если $x_0 > 0$, и нарушению срока выполнения разработки, если $x_0 < 0$.

Предположим, что осталось выполнить самый последний этап с номером 1 и пусть:

- x - величина резерва времени (возможно, и отрицательного);
- a_1 - нормальная продолжительность выполнения этого этапа;
- d_1 - величина, на которую в среднем удлиняется продолжительность этапа за счет

случайных факторов;

- z_1 - выбираемая величина, на которую сокращается продолжительность выполнения этапа.

При этих обозначениях величина запаздывания или опережения заданного срока выполнения разработки x_0 определяется, согласно формуле (2) следующим образом:

$$x_0 = x - a_1 - d_1 + z_1.$$

Найдем минимальные дополнительные затраты на выполнение данного этапа при известном значении величины x . Они определяются величиной z_1 и тем, какого значения при заданных величинах x, a_1, d_1 достигла величина x_0 . Эти затраты можно вычислить по следующей формуле:

$$f_1(x) = \min_{0 \leq z_1 \leq A_1} \{C_1(z_1) + \psi(x_0)\}, \quad (3)$$

$$\text{где } \psi(x_0) = \begin{cases} P(x_0), & x_0 < 0, \\ 0, & x_0 = 0, \\ K(x), & x_0 > 0. \end{cases}$$

В фигурных скобках выражения (3) первое слагаемое – это затраты на сокращение продолжительности выполнения этапа на величину z_1 , причем $0 \leq z_1 \leq A_1$, где A_1 - максимально возможное сокращение продолжительности выполнения этого этапа. Второе слагаемое в фигурных скобках выражения (3) – возможные потери за счет запаздывания срока выполнения разработки, если $x_0 < 0$, или за счет слишком раннего ее окончания, если $x_0 > 0$.

Пусть теперь осталось выполнить два последних этапа, т.е. первый и второй, причем перед их выполнением имеется резерв времени x , возможно, и отрицательный. В этом случае величина запаздывания определяется следующим выражением:

$$x_0 = x - a_2 - d_2 + z_2 - a_1 - d_1 + z_1. \quad (4)$$

Найдем минимальные дополнительные затраты на выполнение этих двух этапов при

известном значении x . Они определяются следующей очевидной формулой:

$$f_2(x) = \min_{\substack{0 \leq z_1 \leq A_1 \\ 0 \leq z_2 \leq A_2}} \{C_1(z_1) + C_2(z_2) + \psi(x_0)\}. \quad (5)$$

В выражении (5) величина x_0 вычисляется уже по формуле (4).

Перепишем выражение (5) в следующем виде:

$$\begin{aligned} f_2(x) &= \min_{\substack{0 \leq z_1 \leq A_1 \\ 0 \leq z_2 \leq A_2}} \{C_1(z_1) + C_2(z_2) + \psi(x_0 = x - a_2 - d_2 + z_2 - a_1 - d_1 + z_1)\} = \\ &= \min_{0 \leq z_2 \leq A_2} \left\{ C_2(z_2) + \min_{0 \leq z_1 \leq A_1} \{C_1(z_1) + \psi(x - a_2 - d_2 + z_2 - a_1 - d_1 + z_1)\} \right\} = \\ &= \min_{0 \leq z_2 \leq A_2} \left\{ C_2(z_2) + \min_{0 \leq z_1 \leq A_1} \{C_1(z_1) + \psi(y - a_1 - d_1 + z_1)\} \right\} \end{aligned} \quad (6)$$

где $y = x - a_2 - d_2 + z_2$.

Из выражений (6) и (3) следует, что выражение (6) можно записать в следующем образом:

$$f_2(x) = \min_{0 \leq z_2 \leq A_2} \{C_2(z_2) + f_1(x - a_2 - d_2 + z_2)\}. \quad (7)$$

Рассуждая аналогично, можно получить следующее выражение для минимальных дополнительных затрат на выполнение трех, четырех и т.д. этапов при условии, что перед началом их выполнения имеется положительный или отрицательный резерв времени x :

$$f_n(x) = \min_{0 \leq z_n \leq A_n} \{C_n(z_n) + f_{n-1}(x - a_n - d_n + z_n)\}. \quad (8)$$

При построении рекуррентных функций (8) для $n = 1, 2, \dots, N$ необходимо фиксировать для каждого значения x то значение $z_n^*(x)$, $n = \overline{1, N}$, при котором функционал, стоящий в фигурных скобках в выражении (8), достигает минимального значения. Полученные по рекуррентным формулам (8) функции $f_n(x)$ и $z_n^*(x)$, $n = \overline{1, N}$ позволяют получить решение поставленной задачи определения значений $z_1^*, z_2^*, \dots, z_N^*$, которые минимизируют ожидаемые затраты на выполнение разработки при заданной директивной продолжительности ее выполнения t .

8.3.3. Сетевые модели выполнения разработок

Одним из широко известных в свое время моделей хода выполнения разработок были так называемые сетевые модели. Эти модели предполагали представление процесса выполнения разработки в виде направленного нагруженного графа, дуги которого – это выполняемые работы (процедуры), а вершины – начало и окончание соответствующих работ (процедур).

Если сетевая модель описывает процесс выполнения какой-то конкретной разработки, то каждая дуга такого графа обычно бывает нагружена наименованием соответствующей работы, продолжительностью выполнения этой работы, видом требуемого для ее выполнения ресурса, в общем случае в виде вектора, и т.д. Кроме того, для каждой вершины графа определены календарные сроки начала или окончания соответствующей работы, для которой эта вершина является либо ее началом, либо ее окончанием. Пример фрагмента такого графа приведен на рис. 8.3.

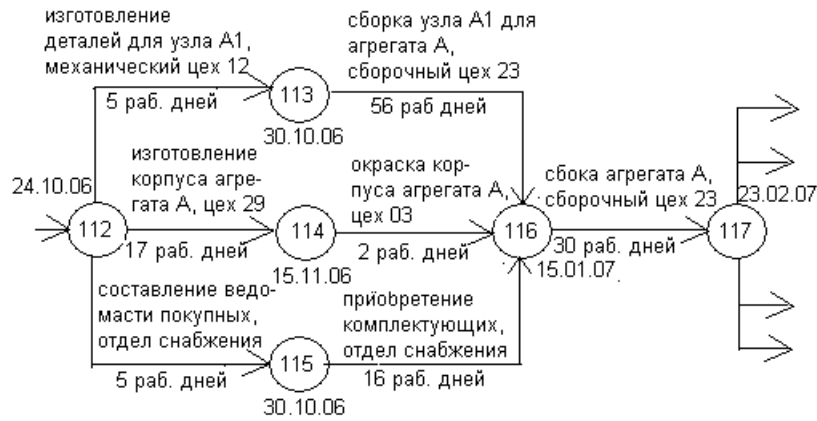


Рис. 8.3. Фрагмент сетевого графика изготовления агрегата А некоторого изделия

19. Модели оперативного управления разработками: модель определения начала выполнения новой разработки и периодичности контроля процесса выполнения разработок.

. Модель определения начала выполнения новой разработки

Одной из особенностей большинства РП является поступление заданий на новые разработки в течение нового года. Это определяет необходимость оперативной корректировки годового тематического плана при составлении квартальных тематических планов, учитывая в них вновь поступившие на предприятие работы.

При поступлении новой разработки, прежде всего, составляется исходный план ее выполнения. Он содержит желаемые сроки выполнения этапов и всей разработки в целом и требуемые для этого ресурсы без учета реальных возможностей подразделений предприятия, которые м.б. в это же самое время заняты работами, ранее включенными в тематический план. Поэтому возникают две взаимосвязанные задачи. Первая задача - определение возможности выполнения новой разработки в сроки, указанные в исходном плане. Если ее решение дает отрицательный результат, то рассматривают вторую задачу - назначают наиболее ранний срок начала выполнения разработки, который определяется, во-первых, занятостью подразделений выполнением работ, ранее включенных в план, а во-вторых, объемом и видом ресурсов, потребных для выполнения новой разработки.

Особенностью процесса каждой разработки является его стохастичность, выражающаяся в неопределенности сроков ее выполнения и требуемых объемов ресурсов. Поэтому модель рассматриваемой задачи должна быть стохастичной.

По смыслу задачи срок начала выполнения новой разработки д.б. по возможности наиболее ранним. Поэтому в качестве критерия оптимизации решения задачи естественно взять следующий критерий:

$$t_{\text{НОВ}}^{\text{НАЧ}} \rightarrow \min, \quad (1)$$

где $t_{\text{НОВ}}^{\text{НАЧ}}$ - срок начала новой разработки, отсчитываемый от данного текущего момента времени решения задачи. Возможные значения искомой величины $t_{\text{НОВ}}^{\text{НАЧ}}$ лежат в пределах от 1 до T , где T - плановый горизонт. Так как на всем интервале времени от $t_{\text{НОВ}}^{\text{НАЧ}}$ до $t_{\text{НОВ}}^{\text{НАЧ}} + \tau - 1$, где τ - продолжительность выполнения новой работы, превышение требуемого количества ресурсов над имеющимися может быть только с некоторой заданной вероятностью, то в качестве ограничения задачи естественно взять следующее:

$$P[Y_k(t) \leq R_k(t)] \geq P_k^d, t = t_{\text{нов}}^{\text{нач}}, t_{\text{нов}}^{\text{нач}} + 1, \dots, t_{\text{нов}}^{\text{нач}} + \tau - 1, \quad (2)$$

где:

$R_k(t)$ -ресурсные возможности предприятия по k -му виду ресурса ($k = \overline{1, K}$) в t -й дискретный интервал времени ($t = \overline{1, T}$);

$Y_k(t)$ -суммарный объем работ, необходимый для выполнения всех разработок данного технического направления, в том числе и новой, в t -м интервале времени;

P_k^d - допустимая вероятность того, что количество ресурсов k -го вида, необходимое для выполнения всех работ данного технического направления, не превысит имеющихся ресурсов в распоряжении предприятия.

Поставленную задачу можно решить, если использовать следующий алгоритм:

1. Значение t_{HOV}^{HAY} полагается равным 1.
2. Для всех $t = t_{HOV}^{HAY}, t_{HOV}^{HAY} + 1, \dots, (t_{HOV}^{HAY} + \tau - 1)$, и $k=1, 2, \dots, K$ проверяется выполнение условия (2). Если оно не выполняется хотя бы для одного t или k , то переходим к выполнению п.3. В противном случае переходим к п.4.
3. Значение t_{HOV}^{HAY} увеличивается на единицу. Если оно превысило величину T , то разработка не может быть начата в заданном плановом горизонте. Если новое значение t_{HOV}^{HAY} не превосходит T , то переходим к п.2.
4. Проверяется возможность выполнения новой работы в установленные сроки. Для этого проверяется выполнение неравенства

$$P[(t_{HOV}^{HAY} + \tau) \leq T_{HOV}^d] \geq P_{cp}^d, \quad (3)$$

где T_{HOV}^d - установленный срок окончания новой работы; P_{cp}^d - заданная вероятность выполнения работы в установленный срок.

Если неравенство выполняется, то переходим к п.5. В противном случае выполняем п.3.

5. На печать выдаются сообщения о положительном решении задачи. После этого работа алгоритма прекращается.

8.5.2. Постановка и вероятностная модель для определения периодичности контроля процесса выполнения разработок

В процессе оперативного управления разработками возникает задача определения оптимальной периодичности опроса исполнителей с целью контроля за ходом выполнения запланированных им работ. Существуют различные подходы к решению этой задачи.

В [11] предложен метод определения частоты опроса исполнителей при выполнении ими одной достаточно крупной работы. Его особенность состоит в следующем.

1. Чем ближе работа к завершению, тем чаще осуществляется опрос исполнителей, т.е. периодичность контроля за ее выполнением различна.
2. Чем лучше идет работа, тем реже осуществляется ее контроль. Наоборот, при отставании ее выполнения частота контроля увеличивается.

Основными недостатками этого метода являются, во-первых, переменная частота контроля, что усложняет его организацию в АСУ РП, а во-вторых, на некотором шаге контроля, в том числе и первом, результат контроля может быть таким, что даже при принятии оперативных решений выполнить работу в установленный срок из-за большого его отставания окажется уже невозможным. Кроме того, метод предполагает контроль за выполнением только одной работы, что не всегда соответствует реальным процессам проведения разработок в подразделениях РП.

Рассмотрим другой подход к решению задачи определения частоты опроса исполнителей, предполагающий, что за подразделением РП закреплен комплекс запланированных ему работ [12].

Пусть этот комплекс состоит из n работ, для каждой из которых задана ее плановая продолжительность b_r и срок начала выполнения a_r , где $r = \overline{1, n}$. План работы подразделения при этих обозначениях можно представить в виде вектора $X = \{a_r, b_r\}, r=1, n$.

За начальный момент времени $t=0$ примем момент начала работы подразделения по составленному плану. В случайные последовательные моменты времени $t=t_1, t_2, \dots, t_i, \dots$ на подразделение действуют возмущения в виде внеплановых работ, болезней исполнителей, отвлечения исполнителей на выполнение других работ и т.п. Эти возмущения в конечном итоге оказывают влияние на длительность выполнения запланированных работ. Обозначим через Δb_r ожидаемое изменение длительности выполнения r -й работы по сравнению с ранее установленной в плане, а через $A(t_i)$ – множество номеров тех работ, для которых в момент $t=t_i$ произошло это изменение.

Ввиду случайности появления возмущающих факторов в момент t_i и их случайного воздействия на длительность выполнения работ из $A(t_i)$ величины $\Delta b_r (r \in A(t_i))$ будут случайными.

При этих обозначениях величина отклонения фактического хода выполнения от запланированного $S(t)$ в момент времени $t>0$ может быть определена следующим образом:

$$S(t) = \sum_{t_i < t} \sum_{r \in A(t_i)} |\Delta b_r|. \quad (1)$$

Являясь суммой случайных величин, заданных в случайные моменты времени $t_1, t_2, \dots, t_i, < t$, эта величина представляет собой дискретный случайный процесс. Обозначим через $F(t, y)$ функцию распределения сечения этого случайного процесса в момент t . Тогда необходимую периодичность контроля за ходом выполнения плана подразделения τ можно определить исходя из условия, что в момент времени $t = \tau$ величина $S(t)$ будет меньше некоторой допустимой величины Δ , при которой с некоторой заданной вероятностью P^d еще можно путем принятия оперативных решений устранить возникшие отклонения. Это условие записывается следующим образом:

$$P(S(\tau) < \Delta) = F(\tau, \Delta) = P^d. \quad (2)$$

Рассмотрим, как можно найти функцию распределения $F(t, y)$, воспользовавшись рядом оправданных для практики работы РП допущений.

Пусть $v(t)$ – число возмущений, которые действовали на подразделение за время t , а $P\{S(t) < y / v(t) = k\}$ – вероятность того, что величина отклонения $S(t)$ в момент t меньше некоторой величины y при условии, что за время t на подразделение действовало ровно k возмущений. По формуле полной вероятности имеем

$$P(S(t) < y) = F(t, y) = \sum_{k=1}^{\infty} P\{S(t) < y / v(t) = k\} P(v(t) = k), \quad (3)$$

где $P(v(t) = k)$ – вероятность, что за время t на подразделение действовало ровно k возмущений.

Сделаем следующее допущения:

1. Случайные величины $S(t) = \sum_{r \in A(t_i)} |\Delta b_r|$ независимы, т.е. величина отклонения плана в момент

времени $t=t_i$ никак не влияет на величину его отклонения в последующие моменты времени $t=t_{i1}, t_{i2}, \dots$.

2. Случайные величины $S(t_i)$ подчиняются одному и тому же показательному закону распределения с известным параметром a , где $\frac{1}{a}$ – среднее значение величины отклонения хода

выполнения плана от номинального значения.

3. Процесс поступления возмущений, влияющих на выполнение плана, - простейший пуассоновский с известным параметром λ , где λ – среднее число возмущений в единицу времени.

При этих допущениях имеем

$$P(v(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad (4)$$

$$P\{S(t) < y / v(t) = k\} = \int_0^y \frac{a^k z^{k-1} e^{-az}}{(k-1)!} dz. \quad (5)$$

Выражение (5) – это функция распределения случайной величины, полученной путем сложения независимых случайных величин, каждая из которых подчинена показательному закону распределения с параметром a .

Подставляя (4) и (5) в (3), получаем выражение, определяющее функцию $F(t, y)$:

$$F(t, y) = \sum_{k=1}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} \int_0^y \frac{a^k z^{k-1} e^{-az}}{(k-1)!} dz. \quad (6)$$

Выражение (6) можно записать в виде [12]

$$F(t, y) = e^{-\lambda t} F_0(y) + e^{-\lambda t} \sqrt{\lambda t a} \int_0^y \frac{e^{-az}}{\sqrt{z}} I_1(2\sqrt{\lambda t a z}) dz, \quad (7)$$

где $F_0(y)$ – вероятность отклонения плана подразделения на величину, не превосходящую при условии, что за время t не действовали никакие возмущения; $I_1(u)$ – функция Бесселя чисто мнимого аргумента первого рода.

Очевидно, что

$$F_0(y) = \begin{cases} 1, & \text{если } y > 0, \\ 0, & \text{если } y \leq 0. \end{cases} \quad (8)$$

Из выражений (2) и (7) следует, что для нахождения искомой периодичности контроля хода выполнения плана работы подразделения τ при известных значениях величин a , τ и Δ необходимо решить следующее выражение:

$$e^{-\lambda t} F_0(\Delta) + e^{-\lambda t} \sqrt{\lambda t a} \int_0^{\Delta} \frac{e^{-az}}{\sqrt{z}} I_1(2\sqrt{\lambda t a z}) dz = P^d. \quad (9)$$

Так как $\Delta > 0$, то, воспользовавшись выражением (8), вместо (9) можно записать следующее:

$$\sqrt{\lambda t a} \int_0^{\Delta} \frac{e^{-az}}{\sqrt{z}} I_1(2\sqrt{\lambda t a z}) dz = P^d e^{\lambda \tau} - 1. \quad (10)$$

Уравнение (10) можно решить одним из известных численных методов интегрирования.

20. Назначение унифицированного языка моделирования UML

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

21. Виды унифицированного языка моделирования UML.

UML 1.5 определял двенадцать типов диаграмм, разделенных на три группы:

- четыре типа диаграмм представляют статическую структуру приложения;
- пять представляют поведенческие аспекты системы;
- три представляют физические аспекты функционирования системы (диаграммы реализации).

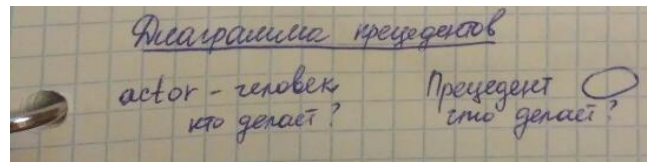
Итак, мы кратко рассмотрим такие виды диаграмм, как:

- *диаграмма прецедентов;*
- *диаграмма классов;*
- *диаграмма объектов;*
- *диаграмма последовательностей;*
- *диаграмма взаимодействия;*
- *диаграмма состояний;*
- *диаграмма активности;*
- *диаграмма развертывания.*

22. Диаграммы прецедентов.

Элементы:

1. **Границы системы** (контекст/система) — прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации. Этот элемент диаграммы показывает границу между тем, что вы как *аналитик* показали в виде прецедентов (внутри этих рамок), и тем, что вы изобразили как действующие лица (вне их) - *внешние сущности*, взаимодействующие с моделируемой системой.

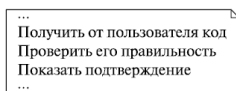


2. **Эктор (actor)** - это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Эктором может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности. Экторы "общаются" с системой через прецеденты.

3. **Прецедент (use-case)** - описание отдельного аспекта поведения системы с точки зрения пользователя. Прецедент не показывает, "как" достигается некоторый результат, а только "что" именно выполняется.

Изображаются прецеденты в виде эллипса, внутрь контура которого помещается имя (описание) прецедента (название сценария). Стрелка - это *направленная ассоциация* и стрелка (как и на других диаграммах) всегда направлена в сторону той сущности, от которой что-то требуют.

Сценарий - это повествовательный рассказ о совершаемых эктором действиях, история, эпизод, происходящий в данных временных рамках и данном контексте взаимодействия. Сценарии также иногда можно увидеть на диаграмме прецедентов.



Иногда их изображают в виде "листа бумаги", на котором написано *имя файла*, - прямоугольника с загнутым нижним левым уголком. В этом случае указанный *файл* содержит в себе описание данного сценария. А иногда *сценарий* записывается в комментарий-прямоугольниками с загнутым верхним правым углом и соединяются с элементом, который они поясняют, пунктирной линией. Сценарий может быть написан в виде *псевдокод*, *таблица*, *диаграмма активностей*,

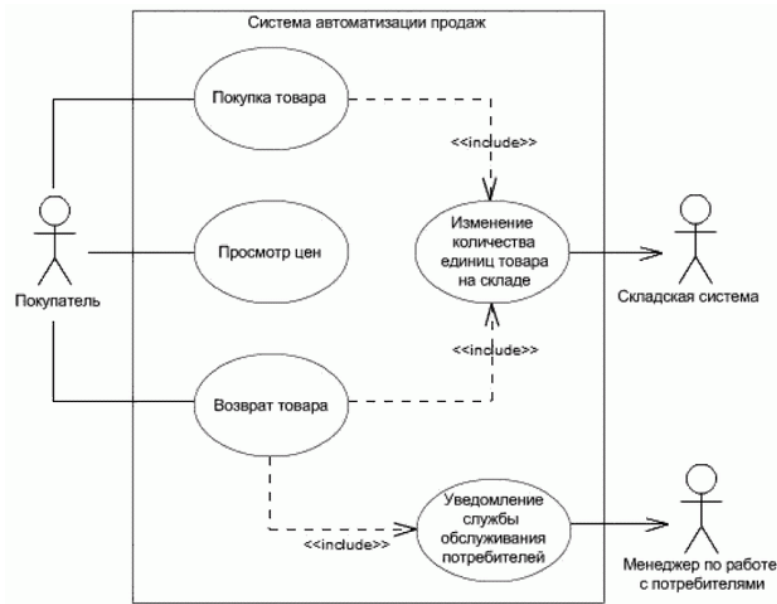
23. Отношения между прецедентами

Часть дублирующейся информации в модели прецедентов можно устранить указанием связей между прецедентами^[1]:

- **обобщение прецедента** (отношение между предком и потомком) — стрелка с незакрашенным треугольником (треугольник ставится у более общего прецедента), *прецедент-потомок* наследует поведение и семантику прецедента-родителя и дополняет его поведение.



- **включение прецедента** — пунктирная стрелка в сторону включаемого прецедента со стереотипом «include», означает, что в некоторой точке базового прецедента содержится поведение другого прецедента. базовый прецедент как бы заимствует поведение включаемых, раскладываясь на более простые прецеденты. Включаемые пр. сами по себе не существуют



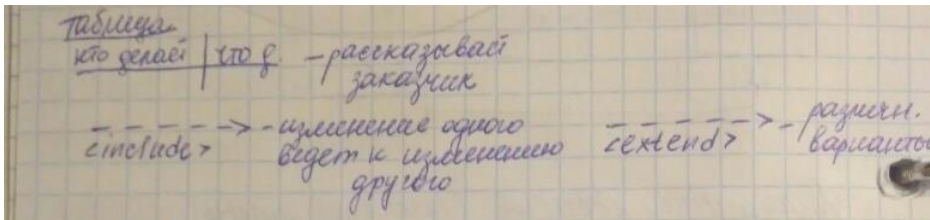
- **расширение прецедента** — пунктирная стрелка со стереотипом «extend» (стрелка входит в расширяемый прецедент, в дополнительном разделе которого может быть указана точка расширения и, возможно в виде комментария, условие расширения) *Отношение* расширения прецедента А к прецеденту В означает, что экземпляр прецедента В может включать в себя (при определенных условиях, которые могут быть описаны в расширении; как именно описаны, мы скажем чуть позже) поведение, описанное в прецеденте А. когда исходный прецедент (а именно, последовательность действий, содержащаяся в нем) приходит в точку расширения, происходит оценка условий расширения. Если условия выполняются, прецедент включает в себя последовательность действий из расширяющего прецедента.



Единственное допустимое отношение между экторами - генерализация (наследование).

ПРИМЕР::





цели создания диаграмм прецедентов:

- определение границы и контекста моделируемой предметной области на ранних этапах проектирования;
- формирование общих требований к поведению проектируемой системы;
- разработка концептуальной модели системы для ее последующей детализации;
- подготовка документации для взаимодействия с заказчиками и пользователями системы.
- Прецеденты дают возможность аналитикам, пользователям и разработчикам говорить на одном языке:
- Прецеденты позволяют разработчикам понять назначение элемента и являются основой для тестирования элемента в течение разработки

ПРИМЕР(дополнительно): Секретарь размещает на сервере меню обеденных блюд на неделю. Сотрудники должны иметь возможность ознакомиться с меню и сделать заказ, выбрав блюда на каждый день следующей недели. Офис-менеджер должен иметь возможность сформировать счет и оплатить его. Система должна быть написана на *ASP.NET*. Такое вот нехитрое интернет-приложение для автоматизации заказов обедов в офис. **Таблица с описанием требований**

Прецедент	Действующее лицо
разместить меню	секретарь
ознакомиться с меню	сотрудник, секретарь, офис-менеджер
сделать заказ	сотрудник, секретарь, офис-менеджер
сформировать счет	офис-менеджер
оплатить счет	офис-менеджер

Здесь нигде не сказано о том, что система должна быть написана на *ASP.NET*. Почему - понятно: это ведь нефункциональное требование - описание таких свойств системы, как особенности среды и реализации, *производительность, расширяемость, надежность* и т. Д

диаграмма прецедентов:

