##

## Базовые сведения о надежности информационных технологий управления

Содержание

Надежность как ключевой фактор развития и применения информационных технологий в управлении

### Надежность автоматизированных систем управления

#### Основные понятия надежности

#### Показатели надежности АСУ

#### Создание надежной АСУ

#### Общий порядок оценки надежности АСУ

#### Обеспечение надежности разрабатываемой (модернизируемой) АСУ

### Системная надежность компьютерных технологий управления

### Информационные технологии создания надежных систем управления

#### Методология структурного анализа и проектирования

#### Сущность методологии SADT

#### Использование методологии IDEF при проектировании и анализе бизнес - процессов

#### Определение системы и модели при проектировании бизнес - процессов

#### Методологии IDEF

#### Методология IDEF и анализ стоимостных характеристик

#### Примеры использования надежных технологий управления

Программное обеспечение как надежная система технологий управления

Надежность программ. Разные подходы

#### Технологии повышения безошибочности программ

надежность информация технология управление

**Надежность как ключевой фактор развития и применения информационных технологий в управлении**

На протяжении всего периода применения компьютеров и компьютерных систем существует тенденция создания высоконадежных управляющих комплексов, ориентированных на получение и использование информационных ресурсов. Эта тенденция выразилась в мощном процессе создания различных видов автоматизированных систем как встроенных в уникальные объекты информационно-технологических комплексов. Это направление является важнейшим в проведении крупных мероприятий по совершенствованию технической и технологической базы систем управления, а также использовании новых методов организации управления, создания автоматизированных производств, основанных на широком применении современного программно-управляемого технологического оборудования, микропроцессорных управляющих вычислительных средств, роботов и промышленных робототехнических систем, средств автоматизации проектно-кострукторских, технологических, организационных и планово-производственных работ.

В научно-техническом отношении на этом этапе для достижения показателей надежности осуществляется синтез ряда разрозненно развивающихся направлений, таких как АСУП (системы автоматизации организационного управления предприятием), САПР (автоматизация проектирования и конструирования), СЧПУ (системы числового программного управления), АСУ ГПС (автоматизированная система управления гибкими производственными системами).

Применение вычислительной техники и средств автоматизации организационных и технологических процессов достигло такого уровня, что был поставлен вопрос о надежной крупномасштабной системной автоматизации на основе компьютерных систем. В этот период были созданы АСУП на базовых предприятиях ведущих отраслей.

При этом, поскольку проблемы надежности функционирования еще не стали основой разработки АСУ, нельзя было точно ответить, где именно и в какой степени проявится наибольший эффект от внедрения новых информационно-технологических средств — в самой технологии или в областях, связанных с организацией, технологией и управлением производства или с проектно - конструкторской и исследовательской деятельностью.

Ориентация на надежность потребовала изучения специфики автономного развития следующих направлений автоматизации: автоматизация обработки информации — автоматизированные системы управления организационными процессами (АСУП), системы автоматизированного проектирования и конструирования (САПР), автоматизация производства на базе использования технологического оборудования с компьютерным управлением (АСУ ГПС), автоматизированные системы управления технологическими процессами в дискретном производстве (АСУ ТП).

Переход к созданию интегрированных систем поставил ряд сложных проблем, связанных прежде всего с тем, что такие системы должны обеспечивать надежное согласованное функционирование территориально рассредоточенных автоматизированных систем с разными показателями надежности различного функционального назначения, базирующихся на разнородной вычислительной технике и взаимодействующих между собой средствами коммуникации.

Организационной основой информатизации управления стало развитие специализации и кооперации производства, гарантирующее надежность элементов АС и их сопряжение в новых перспективных технологиях, создание интегрированной системы управления, охватывающей все стадии жизненного цикла продукции: от формирования заказа на изделие до его поставки и обслуживания у потребителя.

Создание ИАСУ следует рассматривать как новый этап в надежной информатизации технологий управления, основанный на использовании достижений в создании надежных компонент, накопленного опыта разработки и внедрения надежных и эффективных автономных автоматизированных систем, предназначенных для различных видов производственно-хозяйственной деятельности объектов.

Анализ опыта создания надежных АСУ как этапа на пути к информационным ресурсам, формирование которых возможно лишь в рамках надежных компонентов и системных технологий, позволяет сделать вывод о том, что на их основе сложился начальный рынок относительно надежных технологий информационных компонентов (базы данных, комплексы вычислительной техники, типовые проектные решения). Однако с системных позиций теории информационных ресурсов, этот этап может быть отнесен по качественным результатам к процессам натурных исследований проблемных ситуаций. Тогда информация о состояниях параметров надежности альтернатив, которая концентрировалась и обрабатывалась в АСУ, могла рассматриваться лишь как компонент проектного и исследовательского информационного ресурса.

Принципиальные экономические ограничения деятельности фазы натурных исследований при разрешении проблемных ситуаций привели к естественному сужению этого направления информатизации технологий управления, не поддержанного своевременно экспертно-консалтинговым и модельным методами разрешения проблемных ситуаций. Кроме того, область высокой неопределенности, в которой осуществляются операции натурных исследований, надежные альтернативы, разрешение проблемных ситуаций (ПС) было с помощью АСУ возможно лишь с высоким риском, что снижало их реальную и информационную эффективность (рис. 4.1.).

Динамика развития автоматизированных систем разных классов существенно зависит от степени решения двух важнейших аспектов надежности автоматизированных систем — элементной надежности и системной надежности, включая надежность персонала АС.

Несогласованность решения этих проблем породила своеобразную динамику создания систем, отображающих общие закономерности развития надежности, включающей фазы надежной технологии, элементной и системной надежности и перехода к фазе использования нового поколения элементов.

Рис. 4.1. Схема изменения неопределенности ПС при использовании

АСУ разных классов как средств информатизации

Другое направление информатизации технологий управления связано с формированием экспертных информационных компонентов в целях получения информационных ресурсов. Это направление связано непосредственно с бумом развития экспертных систем (ЭС) в 80-90 гг. прошлого столетия как систем искусственного интеллекта, а также с решением проблем надежного функционирования персонала АС. Экспертные системы направлены на повышение надежности локального системного функционирования персонала АС.

Экспертные компьютерные системы-оболочки создали многомиллиардную нишу рынка средств искусственного интеллекта. Были созданы экспертные системы-оболочки (ГУРУ, Интерэксперт и др.), которые стимулировали создание авторских вариантов ЭС как интеллектуальной собственности. На рынке появились демонстрационные версии ЭС (50 - 100 решающих правил), отдельные промышленные версии.

Однако и здесь игнорирование методов надежного экономического, технологического и организационного синтеза информационных компонентов для формирования информационных ресурсов вело к неизбежному затуханию интенсивности работ этого содержательного, но локального этапа на пути к созданию и использованию информационных ресурсов

.

Рис. 4.2.. Изменения неопределенности ПС в процессе локального применения компьютерных экспертных систем

Локальное применение ЭС без предшествующего натурного исследования вело на начальных этапах к росту неопределенности, отсутствию области конкордации (согласия) экспертных оценок по фиксированным решающим правилам, невозможности корректного осуществления остановки экспертизы и как результат - к деградации и временному сужению ниши рынка ЭС на рынке знаний.

Третье важнейшее направление развития информационных компонентов для формирования информационных ресурсов связано с созданием алгоритмических комплексов, экономико-математических моделей. На территории СНГ это направление связано, прежде всего, с деятельностью Центрального экономико-математического института АН РФ, Института проблем управления (автоматики и телемеханики) АН РФ, ЦНИИТУ и НИИ экономико-математических методов (г. Минск) и других организаций. Активно занимались моделями информатизации ведущие зарубежные центры — практически все университеты, в том числе Гарвардский университет, Массачусетский технологический институт.

Были созданы комплексы впечатляющих, но локальных моделей без должных интерфейсов с системами натурного исследования проблем (планирование эксперимента) и экспертными системами. Созданные локальные модели на локальных примерах демонстрировали адекватность, однако их применение без соблюдения логической последовательности создания других надежных компонентов для формирования информационных ресурсов вело к возрастанию неопределенности на начальных этапах применения моделей, и, в силу естественных экономических и временных ограничений, привело к остановке модельных исследований конкретных ПС еще до их разрешения (рис. 4.3.).

Необходимо отметить, что анализ локального применения трех ненадежных основных информационных компонент для формирования и использования ИР указывает путь и позволяет прогнозировать новый этап информатизации, основанный на интеграции экономических и информационно-технологических предпосылок. Комплексно взаимосвязанное с надежными интерфейсами применение надежных технологий интегрированных информационных систем, обеспечивающих ограниченное, управляемое надежное осуществление натурного исследования ПС, ориентировано на согласованные надежные экспертно - консалтинговые процедуры, согласование и применение надежных моделей.

t доп.

R доп.

Н доп.

Время, ресурсы

Рис. 4.3. Изменение неопределенности ПС в процессе локального
применения компьютерных экспертных систем

В таких ИАСУ надежность достигается за счет точного плана, ограниченного натурного исследования ПС, за счет ЭС с управляемой процедурой оценки (конкордации) точки согласия и применения экспертных правил остановки экспертизы, что исключает катастрофический рост неопределенности из-за неограниченности экспертных суждений и выхода эксперта за область компетенции при разрешении ПС, за счет надежного интерфейса консалтинговой ЭС с адекватным (надежным) моделирующим комплексом. Это позволяет ограничивать начальный рост неопределенности за счет ненадежных операций, осуществляет управляемые правила остановки операций разрешения ПС, определяет точки начала и конца области информационных ресурсов. Создание партнерской системы осуществляет реальное получение и использование ИР для разрешения ПС.

При этом за счет согласованности и надежности операций взаимное влияние комплексов натурного, экспертного и модельного этапов разрешения проблемной ситуации проявляется через синергетический эффект. В этом случае экономический эффект информационных ресурсов связан с акселерацией и мультипликацией в экономических системах. Результаты согласованных взаимодействий интегрированных натурного, экспертного и модельного компонентов ведут к возникновению в результате функционирования АСУ важнейшей экономической категории - информационных ресурсов — субститута другим ресурсам для разрешения ПС.

Таким образом, для успешного разрешения проблемных ситуаций, возникающих в процессе управления, должны быть созданы надежные автоматизированные системы управления, снижающие риск принятия неверных решений до минимума. Динамика развития автоматизированных систем, объем их рынка существенно зависят от успешного решения проблем надежности компонентов и системной надежности, надежных технологий их функционирования.

###

### Надежность автоматизированных систем управления

В данном разделе приведены основные положения по надежности АСУ, номенклатура основных показателей надежности, порядок установления требований к надежности АСУ, общие принципы оценки надежности, состав работ по обеспечению надежности АСУ. Словарь терминов по надежности, встречающихся в данном разделе, приведен в приложении.

#### Основные понятия надежности

Обеспечение необходимого уровня надежности требует проведения специального комплекса работ, выполняемых на разных стадиях создания и эксплуатации систем управления.

При решении вопросов, связанных с обеспечением требуемого уровня надежности АСУ, учитываются следующие особенности:

* каждая АСУ является многофункциональной системой, функции которой имеют существенно различную значимость и, соответственно, характеризуются разным уровнем требований к надежности их выполнения;
* возможно возникновение некоторых исключительных (аварийных, критических) ситуаций, представляющих сочетание отказов или ошибок функционирования системы и способных привести к значительным нарушениям функционирования объекта управления (авариям);
* в функционировании АСУ участвуют различные виды ее обеспечения и персонал, которые могут в той или иной степени влиять на уровень надежности системы управления;
* в состав каждой АСУ входит большое количество разнородных элементов: технических, программных, эргатических и др., при этом в выполнении одной функции АСУ обычно участвуют несколько различных элементов, а один и тот же элемент может участвовать в выполнении нескольких функций системы.

При решении вопросов надежности АСУ количественное описание, анализ, оценку и обеспечение надежности проводят по каждой функции АСУ в отдельности. В необходимых случаях используют также анализ возможности возникновения в системе аварийных ситуаций, ведущих к значительным техническим, экономическим или социальным потерям вследствие аварии объекта управления (или автоматизированного комплекса в целом).

Функции АСУ подразделяют на простые и составные. Для некоторых АСУ возможно построение составной функции наиболее общего вида, отображающей функционирование АСУ в целом.

Перечень функций и видов их отказов, по которым задаются требования к надежности конкретной АСУ, а также критерии этих отказов устанавливает заказчик по согласованию с разработчиком и вносит в техническое задание (ТЗ на АСУ). Для установления критериев отказов составляют перечень признаков или параметров, по которым может быть обнаружен факт возникновения каждого отказа, а при необходимости — количественных (критериальных) значений этих параметров.

Если для некоторой функции АСУ определено несколько видов отказов, существенно различающихся по причинам возникновения или по вызываемым ими последствиям, то безотказность и ремонтопригодность по этой функции задают отдельно по каждому виду отказов. При этом критерии отказов устанавливают по каждому виду отказов.

Перечень рассматриваемых аварийных ситуаций, по которым задают требования к надежности, составляет заказчик по согласованию с разработчиком и вносит в техническое задание с указанием, при каких условиях эксплуатации рассматривают возникновение каждой из приведенных аварийных ситуаций.

Уровень надежности АСУ зависит от надежности и других свойств ее технического обеспечения (комплекса технических средств), программного обеспечения и персонала, участвующего в функционировании АСУ.

Уровень надежности АСУ зависит от следующих основных факторов:

* состава и уровня надежности используемых технических средств, их взаимосвязи в надежностной структуре комплекса технических средств АСУ (КТС АСУ);
* состава и уровня надежности используемых программных средств, их содержания (возможностей) и взаимосвязи в структуре программного обеспечения АСУ (ПО АСУ);
* уровня квалификации персонала, организации работы и уровня надежности действий персонала АСУ; рациональности распределения задач, решаемых системой, между КТС АСУ, ПО АСУ и персоналом АСУ;
* режимов, параметров и организационных форм технической эксплуатации КТС АСУ;
* степени использования различных видов резервирования (структурного, информационного, временного, алгоритмического, функционального);
* степени использования методов и средств технической диагностики; реальных условий функционирования АСУ.

Совокупность технических, программных и эргатических элементов АСУ (технических и программных средств и части персонала АСУ), выделяемая из всего состава АСУ по признаку участия в выполнении некоторой (-й) функции системы, образует -ю функциональную подсистему АСУ (ФП АСУ).[[1]](#footnote-1)

Анализ надежности АСУ в реализации ее функций проводят по каждой ФП АСУ в отдельности с учетом уровня надежности и других свойств входящих в нее технических, программных и эргатических элементов.

Выбор состава показателей надежности АСУ производят на основе установленных техническим заданием перечня функций системы, перечня видов их отказов и перечня аварийных ситуаций, по которым регламентируют требования к надежности.

Требуемые численные значения выбранных показателей надежности АСУ (требования к надежности) устанавливаются по определенным критериям на основе анализа влияния отказов АСУ в выполнении ее функций и аварийных ситуаций на эффективность функционирования автоматизированного комплекса (АСУ и объект управления) в целом, а также затрат, связанных с обеспечением надежности.

Оценку надежности АСУ проводят на различных стадиях создания и эксплуатации АСУ.

При разработке АСУ проводят проектную (априорную) оценку надежности системы. При опытной и промышленной эксплуатации АСУ проводят экспериментальную (апостериорную) оценку надежности системы.

Оценку надежности АСУ производят с учетом надежности КТС АСУ и, при необходимости, с учетом надежности ПО АСУ и действий персонала АСУ. Необходимость учета надежности ПО АСУ и действий персонала АСУ при оценке надежности АСУ на разных стадиях создания и эксплуатации устанавливают техническим заданием на АСУ.

#### Показатели надежности АСУ

В качестве показателей надежностями АСУ используют показатели, характеризующие надежность реализации функций системы и опасность возникновения в системе аварийных ситуаций.

Описание надежности АСУ по функциям (по ФП АСУ) осуществляют:

* по отдельным составляющим надежности единичным показателям;
* по нескольким составляющим надежности совместно комплексным показателям надежности.

Для описания надежности АСУ по непрерывно - выполняемым функциям (Н-функции) и по дискретно - выполняемым функциям
(Д-функции) используют различные показатели.

Описание безотказности и ремонтопригодности АСУ по
Н-функциям осуществляют с помощью единичных или комплексных показателей надежности.

Основными единичными показателями безотказности являются:

* средняя наработка системы на отказ в выполнении -й функции (средняя наработка на отказ -й ФП АСУ) -;

* вероятность безотказного выполнения системой -й функции (вероятность безотказной работы -й ФП АСУ) в течение заданного времени .

Допускается использовать следующие показатели:

* среднюю наработку системы до отказа в выполнении -й функции (средняя наработка до отказа -й ФП АСУ) - ;

* параметр потока отказов системы в выполнении -й функции (параметр потока отказов -й ФП АСУ) - ;

* интенсивность отказов системы в выполнении -й функции (интенсивность отказов -й ФП АСУ) - .

Основным и единичными показателями ремонтопригодности являются:

* среднее время восстановления способности системы к выполнению -й функции после отказа (среднее время восстановления -й ФП АСУ) ;

* вероятность восстановления в течение заданного времени способности системы к выполнению -й функции после отказа (вероятность восстановления -й ФП АСУ за время ) - .

Комплексными показателями безотказности и ремонтопригодности являются:

* коэффициент готовности системы к выполнению -й функции (коэффициент готовности -й ФП АСУ) - ;

* коэффициент технического использования системы по -й функции (коэффициент технического использования -й ФП АСУ) – *K TC i ;*

* коэффициент сохранения эффективности системы по -й функции (коэффициент сохранения эффективности -й ФП АСУ) -.*К ЭФ i.*

Описание безотказности и ремонтопригодности АСУ по
Д-функции осуществляют с помощью комплексных показателей надежности.

Основным комплексным показателем безотказности и ремонтопригодности системы в отношении выполнения ею -й
Д-функции является вероятность успешного выполнения системой заданной процедуры при поступлении запроса (вероятность успешного выполнения заданной процедуры -й функциональной подсистемой АСУ) - .

Дополнительным комплексным показателем безотказности и ремонтопригодности системы в отношении выполнения ею -й Д-функции является вероятность успешного выполнения и последовательно поступающих запросов (n).

Описание надежности АСУ по аварийным ситуациям осуществляют с помощью комплексных показателей надежности.

Показателями надежности АСУ по аварийным ситуациям являются показатели, характеризующие:

* опасность возникновения аварийной ситуации в течение некоторого заданного интервала времени нормального функционирования системы;
* опасность возникновения аварийной ситуации в результате воздействия на систему внешнего экстремального фактора.

Для описания надежности АСУ по аварийным ситуациям могут быть использованы следующие показатели:

* средняя наработка системы до возникновения в ней -й аварийной ситуации при нормальных условиях функционирования АСУ - ;

* вероятность возникновения в системе -й аварийной ситуации в течение заданного времени при нормальных условиях функционирования АСУ - ;

* вероятность возникновения в системе -й аварийной ситуации в результате воздействия -го экстремального воздействующего фактора .

Допускается также использование следующих показателей:

* вероятность отсутствия (невозникновения) в системе -й аварийной ситуации в течение заданного времени при нормальных условиях функционирования АСУ - ;

* вероятность отсутствия (невозникновения) в системе -й аварийной ситуации в результате воздействия -го экстремального воздействующего фактора .

Описание долговечности АСУ осуществляют по АСУ в целом или, при необходимости, по отдельным ее подсистемам с помощью единичных показателей надежности.

Основными показателями долговечности являются:

* средний ресурс -й подсистемы АСУ (АСУ в целом) - ;

* средний срок службы -й подсистемы АСУ (АСУ в целом) - .

Допускается также использовать следующие показатели:

* гамма-процентный ресурс -й подсистемы АСУ (АСУ в целом) —;

* + гамма-процентный срок службы -й подсистемы АСУ (АСУ в целом) — .

В обоснованных случаях, кроме показателей надежности АСУ, допускается использовать показатели, установленные ГОСТ 27.002 83,
ГОСТ 13216 74, ГОСТ 21623 76.

#### Создание надежной АСУ

Установление требований к надежности конкретной разрабатываемой АСУ состоит в выборе состава (номенклатуры) показателей, используемых для количественного описания надежностных свойств системы, и определении требуемых числовых значений (норм) этих показателей.

Показатели надежности вводят по каждой функции системы и по каждому виду их отказов, а также по установленным для рассматриваемой системы аварийным ситуациям.

Состав показателей надежности определяют на основе включенных в ТЗ на АСУ перечней функций, видов их отказов и тех аварийных ситуаций, для которых следует устанавливать требования к надежности.

Для каждой из указанных в ТЗ на АСУ функций и по видам их отказов вводят показатели безотказности и ремонтопригодности.

Для каждой из указанных аварийных ситуаций вводят показатели надежности.

Показатели долговечности вводят, при необходимости, для АСУ в целом либо для отдельных ее подсистем в случаях, если по условиям функционирования системы, или по иным причинам, ремонт или замена некоторых технических средств, необходимых для выполнения функций системы и отказавших или выработавших свой ресурс либо срок службы, невозможна без капитального или среднего ремонта или без реконструкции системы. Необходимость установления показателей долговечности указывают в ТЗ на АСУ.

Определение требуемых численных значений введенных показателей надежности АСУ осуществляют по заданным критериям.

Исходными данными для определения обоснованных требований к надежности АСУ являются:

* виды и критерии отказов по всем рассматриваемым функциям системы;
* уровень эффективности по всем функциям системы и величины ущербов по всем видам отказов;
* состав технических, программных и эргатических элементов, участвующих в выполнении каждой функции системы;
* возможные пути повышения надежности для каждой ФП АСУ и связанные с ними затраты;
* величины ущербов, связанных с возникновением возможных в АСУ аварийных ситуаций;
* возможные пути снижения опасности возникновения аварийных ситуаций и связанные с ними затраты.

Требования к надежности АСУ определяют в основном путем сопоставления потерь, связанных с отказами АСУ в выполнении функций и возникновением аварийных ситуаций, и затрат, связанных с обеспечением и повышением надежности АСУ (включая удорожание обслуживания).

Требования к надежности АСУ устанавливают по согласованию между разработчиком и заказчиком АСУ при разработке ТЗ на АСУ.

#### Общий порядок оценки надежности АСУ

Оценку надежности АСУ (по функциям и по аварийным ситуациям) проводят:

* при разработке системы с целью прогноза ожидаемого уровня надежности АСУ (проектная, априорная оценка);
* при вводе системы в эксплуатацию и в процессе ее функционирования с целью определения фактически достигнутого уровня надежности АСУ и проверки его соответствия требованиям к надежности, установленным в ТЗ на АСУ (экспериментальная, апостериорная оценка).

Проектную оценку надежности АСУ, в зависимости от особенностей системы и стадии ее создания, проводят с учетом свойств:

* только комплекса технических средств АСУ;
* КТС АСУ и программного обеспечения АСУ;
* КТС АСУ и персонала АСУ;
* КТС АСУ, ПО АСУ и персонала АСУ.

Проектная оценка надежности АСУ с учетом только КТС АСУ, проводимая на начальных этапах разработки системы, является ориентировочной и ее используют для предварительного определения состава и структуры КТС АСУ.

Проектная оценка надежности АСУ с учетом КТС АСУ и персонала, проводимая при разработке эскизного проекта системы, является ориентировочной, и ее используют для определения целесообразного уровня автоматизации управления объектом, распределения задач между техническими средствами и персоналом АСУ в выполнении функций системы.

Проектную оценку надежности АСУ с учетом КТС АСУ и ПО АСУ проводят при разработке технического проекта и используют для уточнения состава и структуры КТС АСУ, определения требований к надежности, а также выбора способов повышения надежности функционирования технического и программного обеспечения системы.

Проектная оценка надежности АСУ с учетом КТС АСУ, ПО АСУ и персонала АСУ, приводимая при разработке рабочего проекта системы, является более полной, и ее используют для уточнения состава и структуры КТС АСУ, состава и структуры ПО АСУ, состава и структуры задач персонала АСУ, а также, для уточнения взаимодействия КТС, ПО АСУ и персонала АСУ (компонентов АСУ) в реализации функций системы.

Проектную оценку надежности АСУ допускается проводить следующими методами:

* аналитическими;
* вероятностного моделирования;
* комбинированными, представляющими собой сочетание аналитических методов и методов моделирования;
* экспертными.

Экспериментальная оценка надежности АСУ учитывает совместное (результирующее) воздействие на уровень надежности системы КТС АСУ, ПО АСУ и действий персонала АСУ, а также всех реально воздействующих факторов внешней среды, режимов и параметров технической эксплуатации, режимов функционирования системы, внешних помех.

Экспериментальную оценку надежности АСУ допускается проводить:

* путем организации и проведения специальных испытаний на надежность;
* путем сбора и обработки статистических данных о надежности АСУ в условиях ее опытного и промышленного функционирования;
* комбинированными методами, использующими оба эти направления;
* расчетно-экспериментальными методами.

Необходимость проведения оценок надежности АСУ на различных стадиях ее создания и функционирования, методы получения таких оценок, а также состав оцениваемых при этом показателей надежности АСУ указывается в ТЗ на АСУ. Методы проектной и экспериментальной оценок надежности АСУ на разных стадиях создания и эксплуатации системы выбирают с учетом особенностей конкретной разрабатываемой (модернизируемой) АСУ и конкретных условий ее разработки. Учитывается наличие инженерных методик, алгоритмов и программ решения задач оценки надежности, наличие необходимых исходных данных для использования определенного метода и возможности проведения испытаний необходимого объема и пр. При проведении проектной и экспериментальной (расчетно-экспериментальными методами) оценок надежностями АСУ следует использовать данные по надежности элементов АСУ, приведенные в документации их изготовителей и разработчиков, в официальных отчетах об эксплуатации элементов АСУ, а также в справочниках.

#### Обеспечение надежности разрабатываемой (модернизируемой) АСУ

Необходимый уровень надежности конкретной АСУ обеспечивают специальным комплексом работ, проводимых на всех стадиях создания и функционирования АСУ.

К обязательным работам по обеспечению надежности АСУ, которые следует выполнять в процессе создания любой АСУ, относят:

* анализ состава и содержания функций разрабатываемой (модернизируемой) АСУ, определение конкретного содержания понятия «отказ» и критериев отказа по каждому виду отказов для всех функций системы; анализ, при необходимости, аварийных ситуаций в АСУ, определение конкретного содержания понятия «аварийная ситуация» для данной АСУ и критериев аварийной ситуации по каждой из рассматриваемых ситуаций;
* выбор состава показателей надежности по всем функциям АСУ, указанным в ТЗ на АСУ и, при необходимости, по всем аварийным ситуациям, и определение требований к уровню их значений;
* выбор методов оценки надежности АСУ на различных стадиях ее создания и функционирования;
* проведение проектной оценки надежности АСУ при разработке технического проекта системы;
* определение режимов и параметров технической эксплуатации АСУ.

Состав, содержание и последовательность выполнения работ по обеспечению надежности системы устанавливают в «Программе обеспечения надежности АСУ», которую составляют для каждой вновь разрабатываемой или модернизируемой АСУ с учетом специфики системы и условий ее функционирования, важности выполняемых ею функций, требуемого уровня надежности, общего объема затрат на создание, а также особенностей ее создания (наличия необходимых исходных данных, сведений о надежности систем-аналогов и применяемых элементов и пр.).

Создание компьютерных систем управления должно базироваться на точном анализе возможностей реального построения системы и достижения ее целей. Для этого используют экономико-математические методы и модели, позволяющие создать надежные автоматизированные системы управления.

### Системная надежность компьютерных технологий управления

При создании интегрированных систем автоматизированного управления важна роль оптимизационных расчетов, заключающаяся в определении условий, при которых существует заданное значение целевой функции принимаемое за экстремум. Чтобы считать систему интегрированной относительно заданной функции цели, ограничения должны соответствовать реальным возможностям управления ИАСУ. Характеристикам реализуемости АСУ могут служить показатели надежности. Эти взаимосвязанные значения ограничений образуют базу интеграции.

Комплекс оптимизационных расчетов играет роль системы управления базой интеграции, которая периодически обновляется.

Характеристикой реализуемости АСУП, АСУТП и других видов АСУ могут служить показатели надежности.

База интеграции может содержать разделы, отражающие оптимизационные расчеты показателей надежности АСУП и АСУТП, при которых распределение ресурсов на разработку этих компонентов при проектировании и функционировании, обеспечивает получение заданного эффекта.

Одно из направлений комплексной автоматизации управления связано с созданием организационно-технологических АСУ, относящихся к таким разновидностям интегрированных АСУ (АСУОТ), в которой согласованно функционируют АСУ технологическими процессами (АСУТП) и АСУ организационно-экономического типа (АСУП). В процессе функционирования АСУОТ согласованно решаются объединенные в комплекс задачи АСУП и АСУТП, образующие функциональную часть АСУОТ (рис. 4.4). В этих условиях параметры надежности функционирования элементов АСУП и АСУТП, в качестве которых выступают персонал и средства обеспечивающей части, должны быть согласованы, что определяет необходимость согласованного распределения ресурсов на создание АСУП и АСУТП.

Согласованное распределение ресурсов между различными компонентами автоматизированных систем, например АСУТП и АСУП, ведет к формированию, объединенных единой базой интеграции и системной технологией, организационно-технологических АСУ.

Рис.6 Основные свойства ИАСУ, реализующих общесистемную информационную технологию

Интегрированная автоматизированная система управления на промышленном предприятии

Свойства

Представление ИАСУ как многосвязной системы управления с перекрестными и корректирующими связями, реализующими определенный алгоритм взаимодействия из множества возможных и допустимых

Автономность и совместимость компонент

Многосвязность компонент

Представление ИАСУ как системы, обеспечивающей гарантированное достижение конечных целей управления за счет многослойного и временного согласования целей и переодичности взаимодействия компонент, исследования степени достижения целей согласования параметров надежности элементов компонент, обеспечивающих персоналу необходимую для управления информацию, независимо от возмущений.

Инвариантность

Представление ИАСУ в виде многоуровневой системы, включающей в качестве компонент АСУП, АСУТП, САПР и интерфейс между ними

АСУТП может быть представлена взаимодействующими элементами: “производственный персонал - средства автоматизации - технологическое оборудование”. Компенсацию потока различных нарушений в технологическом процессе обеспечивает система автоматизированного организационного управления (АСУП). Параметры элементов АСУТП и АСУП связаны между собой. Оба вида систем достаточно сложные, многоэлементные с большим числом состояний. Для таких систем и их объединения в виде АСУОТ справедливы соотношения, связывающие энтропию распределения параметров надежности элементов, определяющих состояния систем, вызванных ненадежностью работы персонала и средств АСУОТ.

Анализ совместного функционирования систем технологического и организационного управления позволяет решить важный вопрос стратегии автоматизации, оптимально распределить ресурсы, затрачиваемые на автоматизацию каждой из систем.

Надежность каждого из элементов АСУП и АСУТП характеризуется произведением величины Кг (коэффициент готовности), Ро (вероятность безотказной работы) и Рs (вероятность безошибочной работы).

Те же параметры надежности могут характеризовать работу персонала АСУТП, обслуживающего технологическое оборудование, а также персонала АСУП, обслуживающего организационное управление. Низкое значение произведения величин Кг, Рs, Ро в неавтоматизированной системе определяется значительными информационными ошибками. Взаимодействие персонала с устройствами АСУТП и АСУП направлено на улучшение параметра

R=Kг⋅Po⋅Ps.

Информационные аспекты, связанные с оценкой деятельности персонала АСУП и АСУТП, позволяют оценить составляющие параметра R=Kг⋅Po⋅Ps для персонала и его изменение в условиях автоматизированного управления.

Параметр состояния R в каждой системе (АСУТП или АСУП) определяет вероятность своевременного (Кг, безотказного (Po), безошибочного (Ps) выполнения работ по управлению оборудованием (АСУТП) или управлению организационными процессами (АСУП). Тогда его можно представить в виде:

λ=1,2,...,2m

где m - число элементов одного из типов в данной подсистеме АСУП или АСУТП (персонал, оборудование). Число состояний - 2m. Энтропия[[2]](#footnote-2) распределений элементов подсистемы .

В общем случае параметры Кг, Рs, Ро элементов подсистемы персонала зависимы от параметров элементов подсистемы оборудования АСУТП и АСУП. Это определяет необходимость нахождения энтропии подсистемы персонала как условной энтропии при заданном распределении параметров состояния подсистемы оборудования.

Энтропия - неопределенность состояния систем управления АСУТП или АСУП равна сумме энтропии подсистем персонала и оборудования . Для АСУТП , Для АСУП На=Ну+Нр/у Для АСУОТ НS=НА+hБ.

Схема взаимодействия объединенных в систему управления зависимых подсистем “производственный персонал — технологическое оборудование” (Б), и подсистемы автоматизации организационного управления - управленческий персонал” (А) приведена на рис. 4.5.

КИ контрольные и измерительные устройства

ЛП-логические и программные устройства

У-устройства управления

ТОУ-технологический объект управления

- подсистема автоматизации технологических процессов (А)

- первичная информация

- материальный поток

- управляющая информация

- подсистема автоматизации организационного управления (Б)

Рис.4.5. Состав и структура средств автоматизации технологических процессов и организационного управления

Управляющие воздействия

АСУП

ТОУ

АСУТП

Материальный

Подсистема сбора информации

Подсистема накопления информации

Подсистема переработки информации

**Управление персоналом**

КИ

Изделие

Управляющие воздействия

Производственный персонал

Производственный персонал

Производственный персонал

ЛП

У

У

ЛП

КИ

КИ

ЛП

У

Изделие

Изделие

Оборудование

Оборудование

Оборудование

Общая энтропия системы , где  — условная энтропия подсистемы А при фиксированном уровне энтропии подсистемы Б. Если hБ увеличивается, то очевидным следствием зависимости систем является увеличение потока информации, поступающего управленческому персоналу подсистемы А из-за возмущений в подсистеме Б, и увеличение необходимого объема данных для принятия управленческих решений. Возрастание потока информации при увеличении hБ может быть проиллюстрировано на примере устранения управленческим персоналом средствами организационного управления таких возмущений подсистемы управления технологическими процессами, как необнаруженные нарушения в управляемом объекте. После внедрения комплекса средств автоматизации, позволяющих осуществить идеальное (в соответствии с параметрами базы интеграции) организационное и технологическое управление, управленческому и производственному персоналу, должен быть создан режим работы, свободный как от избытка информации, так и от ее недостатка. Тогда подсистемы А и Б могут рассматриваться как независимые в предположении, что используемые средства автоматизации управления и алгоритмы их работы могут создать персоналу режим, инвариантный к возмущениям определенного вида. В этом случае обозначим энтропию подсистем НN, hN соответственно. Создание промежуточных по качеству подсистем А и Б, не обеспечивающих указанного условия, сохраняет зависимость между параметрами подсистем А и Б.

Сравнение величины .с позволяет выбрать направление автоматизации и решить вопрос об оптимальном распределении ресурсов для создания АСУП и АСУТП в составе АСУОТ при котором обеспечивается максимальный эффект автоматизации. Рассматриваемые связи отражены на рис. 4.6.

Организационно-технологическая интегрированная

общесистемная информационная технология АСУ

Управление независимыми параметрами

отдельного элемента объекта

Функциональная часть

Периодичность управления не зависит от параметров группы элементов объекта

Функции управления,

комплексы задач

Персонал

Управление зависимыми параметрами группы элементов

объекта без изменения параметров элементов

Периодичность управления определяется значениями

параметров группы элементов объекта

Обеспечивающая часть

Данные

(ИО)

Технические

средства (ТО)

Модели

управл. (МО)

Программы

(ПО)

АСУТПП

АСУП

Персонал

Виды обеспечения

Персонал

Виды обеспечения

ИО

МО

ПО

ТО

ИО

МО

ПО

ТО

Рис.4.6. Характеристика взаимодействия элементов и комплекса реализуемых задач

Комплекс задач,

объединяемых на

основе внутривидовой,

внутрифазовой интеграции

Комплекс задач межвидовой

внутрифазовой интеграции

Комплекс задач организационно-технологической интеграции

Комплекс задач межвидовой интеграции, обеспечивающей части на базе совместимых видов обеспечения

Периодич-

ность

Элементы

АСУ

Фазы управления

Управ-

ление

Конт-

роль

Изме-

рение

(У)

(К)

(И)

В результате может быть найдено соотношение величин ресурсов, которые следует направить на автоматизацию технологических процессов и организационного управления.

Изложенный подход может быть положен в основу принимаемых решений на этапах технико-экономического (ТЭО) и технического задания (ТЗ) при создании АСУОТ и других классов интегрированных систем.

В рассмотренном случае база интеграции включает комплекс надежностных характеристик элементов АСУП и комплекс надежностных характеристик элементов АСУТП (Кг, Ро, Рs), при которых энтропия распределения параметров этих систем не превышает заданных величин. Такие характеристики отражают физическую реализуемость АСУОТ.

В случае соответствия требованиям базы интеграции организационно-технологическая АСУ будет интегрированной относительно параметров максимального эффекта Б, превышающего сумму эффектов, которые могут быть получены при локальном функционировании АСУП и АСУТП.

Динамическое обновление допустимых значений параметров АСУП и АСУТП путем установления нормативов надежности в результате решения рассмотренной задачи оптимизации распределения ресурсов обеспечивает обновление (актуализацию) базы интеграции.

Таким образом, важнейшим разделом базы данных интегрированных АСУ (ИАСУ) должна стать база интеграции, формируемая на основе комплекса оптимизационных надежностных расчетов, выполненных на разных этапах разработки и функционирования ИАСУ, отражающих реализуемость (достижимость) целей управления.

Экономико-математические методы и модели не нашли бы широкого применения в практике, если бы не были разработаны и применены надежные методологии создания и организации функционирования прикладных систем обработки данных АСУ как их разновидности. Применение надежных методологий создания АСУ и практические их использование повлияло на повышение продуктивности производства посредством систематического внедрения надежных компьютерных технологий, которое потребовало совершенствования методов, анализа и передачи информации.

### Информационные технологии создания надежных систем управления

В конкретных системах, к которым относится система человек-машина, необходимо оценить элементную надежность и надежность структурных процессов по обработке данных так, чтобы с учетом ненадежности реальных элементов можно было бы осуществлять надежную и устойчивую работу системы. Это может быть достигнуто рациональным структурированием процессов, разделением на процедуры, операции (блоки, дуги) с тем, чтобы в нужных местах проводить резервирование, синхронизировать действия и исключать влияние ненадежных элементов. При этом должен быть обеспечен непрерывный и эффективный поток (Workflow). Это создает условия для формализации процессов надежного функционирования систем управления и упреждающих воздействий менеджера на процессы в управляемых системах путем их моделирования. На основе моделирования проводится реструктуризация процесса, формируются упреждающие действия к функциональному элементу в целях повышения надежности, а элементы, не отвечающие требованиям надежности, выбывают из системы.

В данном разделе эти менеджерские проблемы обеспечения надежного функционирования рассматриваются на примерах широко распространенных технологий структурирования, моделирования основных элементов, а также технологии организации надежных информационных потоков Workflow.

#### Методология структурного анализа и проектирования

Под словом “система” можно понимать совокупность взаимодействующих компонент и взаимосвязей между ними. Мир, в котором мы живем, можно рассматривать как сложную взаимосвязанную совокупность естественных и искусственных систем. Это могут быть достаточно сложные системы (планеты Солнечной системы), системы средней сложности (космический корабль) или сверхсложные системы (системы молекулярных взаимодействий в живых организмах). Искусственные системы, как правило, по своей сложности занимают среднее положение. Например, всемирная телефонная сеть содержит десятки или даже сотни тысяч переключателей, однако количество взаимодействий этих переключателей не идет в сравнение с количеством взаимодействий молекул в небольшом стакане воды. С точки зрения теории систем такие системы рассматриваются как системы средней сложности.

Под термином “моделирование” понимают процесс создания точного описания системы. Особенно трудным является описание системы средней сложности, таких, как системы коммутаций в телефонных сетях, управление аэровоздушными перевозками или движением подводной лодки, сборка автомобилей, челночные космические рейсы, функционирование перерабатывающих предприятий. Эти системы описать достаточно трудно, потому что невозможно перечислить все их компоненты со своими взаимосвязями. Неспособность дать простое описание, обеспечить понимание таких систем делает их проектирование и создание дорогостоящим и трудоемким процессом и повышает степень их ненадежности. С ростом технического прогресса адекватное описание систем становится все более актуальной проблемой.

Для того, чтобы облегчить описание и понимание искусственных систем, попадающих в разряд средней сложности, в 1969 году была создана и опробована на практике методология SADT.

SADT — аббревиатура слов **S**tructured **A**nalysis and **D**esign **T**echnique (Технология структурного анализа и проектирования) . С 1973 года сфера использования этой методологии существенно расширяется для решения задач, связанных с большими системами. Примерами могут служить проектирование телефонных коммуникаций реального времени, автоматизация производства, создание программного обеспечения для командных и управляющих систем, поддержка боеготовности. Она с успехом применялась для описания большого количества сложных искусственных систем из широкого спектра областей (банковское дело, очистка нефти, планирование промышленного производства, системы наведения ракет, организация материально-технического снабжения, методология планирования, технология программирования). Причина такого успеха заключается в том, что SADT является полной методологией для создания описания систем, основанной на концепциях системного моделирования.

#### Сущность методологии SADT

Использование экспертных систем, систем автоматизированного производства постоянно расширяется. Успех этих систем непосредственно зависит от возможности предварить их разработку и внедрение описанием всего комплекса проблем, которые необходимо разрешить, указанием того, какие функции системы должны быть автоматизированы, определением точек интерфейса человек-машина и того, как взаимодействует система со своим окружением. Иными словами, этап проектирования системы является критическим для создания высококачественных систем. Системное проектирование определяет подсистемы, компоненты и способы их соединения, задает ограничения, при которых система должна функционировать, выбирает наиболее эффективное сочетание людей, машин и программного обеспечения для реализации системы. SADT — одна из самых известных и широко используемых систем проектирования. Программное обеспечение телефонных сетей, системные поддержка и диагностика, долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, конфигурация компьютерных систем, обучение персонала, встроенное программное обеспечение для оборонных систем, управление финансами и материально-техническим снабжением — вот некоторые из областей эффективного применения SADT. Широкий спектр областей указывает на универсальность и мощь методологии SADT, что привело к стандартизации и публикации ее части, называемой IDEF0.

#### Использование методологии IDEF при проектировании и анализе бизнес - процессов

Изменение либо нарушение привычных производственных связей, структурная перестройка экономики, ограничения в дополнительных капиталовложениях, развитие информационных технологий - все это требует совершенствования бизнес - процессов (СБП). Концепция СБП включает непрерывное совершенствование существующих бизнес - процессов, их перестройку и управление качеством (стандарт ISO 9000). Необходимо отметить как достоинства, так и недостатки упомянутой методологии: управляется большими транснациональными консалтинговыми компаниями, которые разрабатывают методологии и программные средства для совершенствования бизнес - процессов; они “сильны” в управлении проектами, имеют большой консалтинговый опыт, однако “слабы” в моделировании и анализе; большая часть поступающих на рынок программных продуктов не соответствует требованиям в области моделирования и анализа.

Основные сферы использования СБП:

* Управление информационными потоками компании;
* Программные средства поддержки закупок/поставок и материально-технического снабжения.
* Проектирование одновременно происходящих бизнес-процессов и их взаимодействий.

Принципы системного анализа применяются для реорганизации и анализа работы компании с целью оптимизации уже существующих технологий и операций, “открытия” и “изобретения” принципиально новых возможностей и подходов. Для того, чтобы создать модель компании “как она есть”, используют методологию IDEF. Чтобы получить информацию о возможных путях совершенствования бизнес – процессов используется методология функционально-стоимостного анализа АВС (**A**ctivity-**B**ased **C**osting), а имитационное моделирование используется для выбора наилучшего решения.

#### Определение системы и модели при проектировании бизнес - процессов

Система — это множество взаимодействующих компонент и связи между ними. При проектировании, модификации или проверке работы системы можно работать непосредственно с системой или моделью системы.

Модель — это символическое представление системы, позволяющее получить информацию и ответить на вопросы относительно системы. Для простых систем работа с моделью проще и дешевле, чем с самой системой, для больших и/или сложных систем — это единственный реальный подход, для некоторых систем (например, управление воздушными перевозками) — это единственная возможность.

Парадигма[[3]](#footnote-3) моделирования — множество абстракций, которые позволяют “схватить” и выразить суть моделируемой системы. Парадигма статического моделирования представляет структуру системы, но не ее поведение во времени. Парадигма динамического моделирования представляет как структуру, так и поведение во времени.

Основные принципы моделирования позволяют:

* Четко сформулировать цель моделирования;
* Создавать модель не большего размера, чем это необходимо для ответа на поставленные вопросы.

Модель не должнаотображать все аспектымоделируемой системы, а лишь те которые необходимы для достижения цели моделирования.

Идеальная модель – это модель, которая полностью отвечает цели моделирования при минимальной стоимости ее создания.

Модель, которая соответствует ясно сформулированной цели, использует соответствующую парадигму, базируется на адекватной информации и создана квалифицированными разработчиками, с большой вероятностью будет успешной.

####

#### Методологии IDEF

При моделировании систем, содержащих дискретно выполняемые функции, используется методология IDEF, посредством которой преобразуется некоторая входная информация (объекты) в выходную.

Методологии **IDEF** (**ICAM** **DEF**inition):

* ***ICAM*** (**I**ntegrated **C**omputer-**A**ided **M**anufacturing);
* ***IDEF0 —***методология создания функциональной модели, которая является структурированным изображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции;
* ***lDEF1 —***методология создания информационной модели, которая представляет структуру и семантику информации, необходимой для поддержки функций в производственной среде или системе;
* ***IDEF2 —***методология, позволяющая построить динамическую модель меняющегося во времени поведения функций, информации и ресурсов производственной системы или среды.

Основа IDEF0***–***методSADT, предназначенный для представления функций системы и анализа системных требований.

В терминах IDEF0система представляется в виде комбинации блоков и дуг:

* Функциональные блоки отображают функции системы;
* Дуги представляют множество различных объектов системы. Они связывают блоки вместе и отображают взаимодействия и взаимосвязи между блоками;
* Между объектами и функциями возможны четыре вида отношения: вход*,* управление*,* выходи механизм*.* Каждое из этих отношений изображается дугой, связанной с определенной стороной блока.

Рис. 4.7. Пример взаимодействия объектов и функций

СтруктураIDEF0-модели*:*IDEF0-модели состоят из набора диаграмм, образующих иерархию. Каждая диаграмма обычно содержит
3-5 функциональных блоков и является подробным описанием функционального блока, расположенного на предшествующем уровне иерархии**.** Декомпозиция*:* Каждый функциональный блок IDEF0-диаграммы может быть декомпозирован, т.е. представлен в виде совокупности других взаимосвязанных блоков, детально описывающих исходный блок.

**Более общее представление**

**Более детальное представление**

**Эта диаграмма является “родителем” этой диаграммы**

Рис. 4.8. Декомпозиция функциональных блоков IDEF-диаграммы

####

#### Методология IDEF и анализ стоимостных характеристик

Для представления информации в форме, понятной для персонала фирмы, непосредственно участвующего в бизнес - процессах, распределении накладных расходов в соответствии с детальным просчетом использования ресурсов, подробным представлением о процессах и их влиянием на себестоимость, разработан метод АВС (**A**ctivity-**B**ased **C**osting) как "операционно-ориентированная" альтернатива традиционным финансовым подходам.

Цель создания IDEF-модели для СБП — достичь улучшений в работе системы по показателям стоимости и производительности. АВС — один из методов, позволяющий указать на возможные пути улучшения стоимостных показателей.

IDEF и АВС связаны, потому что они оба рассматривают систему как множество последовательно выполняемых функций, дуги управления и дуги механизмов IDEF соответствуют стоимостным факторам и ресурсам АВС.

Связь IDЕF и АВС базируется на трех принципах:

* Функция может иметь число, ассоциированное с ней, которое представляет стоимость выполнения этой функции;
* Стоимость функции, которая не имеет декомпозиции,[[4]](#footnote-4) определяется разработчиком модели;
* Стоимость функции, которая имеет декомпозицию, определяется как сумма стоимостей всех функций на ее странице декомпозиции.

Анализ рабочих потоков — это анализ моделей, которые описывают, каким образом различные документыобрабатываются в организации.

Деятельность организации рассматривается как совокупность взаимосвязанных функций, для выполнения которых требуются ресурсы, такие как персонал, оборудование и т.д.

Оценка моделей рабочих потоков производится на основе отчетов об имитационных экспериментах*,* в которых оценивается производительность процессов, указывается на задержки в их реализациях (т.е. на "узкие места") и неэффективное использование ресурсов.

Интегрированный подход:

1. Сбор информации о функциях*,* участвующих в процессе, их взаимодействиях,об использовании ими ресурсови других параметрах, влияющих на производительность, таких как время и стоимость.

2. Сбор информации относительно ресурсов, используемых в процессе, которая может касаться, например, скорости работы (для оборудования) и их стоимости.

3. Создание функционально-ориентированной графической модели процесса с использованием методологии IDEF0 *(Design/IDEF)*.

4. Трансляция IDEF0-модели в модель цветных сетей Петри
(СР - модель) и получение выполняемой имитационной модели.

Имитационные модели создаются путем импортирования IDEF-диаграмм в иерархическую структуру СР - сетей

5. Использование СР - модели для генерации отчетов об имитационных экспериментах и анализа альтернативных решений.

СР - сети предоставляют язык для описания локальных ситуаций, которые не были определены в IDEF0-модели.

Использование цветных сетей Петри

При проектировании и анализе бизнес - процессов используются цветные сети Петри, так как они позволяют накладывать IDEF-диаграммы на СР - сети с сохранением их первоначальной структуры, имеют графическое представление информации и иерархический подход. Формальное представление параллельных процессов позволяет понять, как работает СР - модель, не вникая детально в реализацию средств моделирования, используемых для ее выполнения

Свойство выполняемости обеспечивает возможность создания имитационных моделей, которые могут быть использованы для анализа производительности и возможных путей реализации системы.

Для понимания функций IDEF служит следующий принцип:

Что6ы перейти к динамической модели необходимо интерпретировать IDEF-модель.

Для документов определяются атрибуты:идентификатор*,* тип*,* размер*,* время поступления в систему.

Для ресурсов определяются атрибуты: идентификатор*,* тип*,* скорость*,* стоимость*,* начальное и конечное время использования*.*

Для функций определяется продолжительность выполнения*.*

Метки входных дуг определяют тип входных документов. Метки дуг механизмов определяют тип ресурсов. Метки выходных дуг определяют: вес, время задержки, тип документов.

При проектировании и анализе бизнес - процессов используется методология IDEF0, которая легко изучаема, поскольку содержит всего лишь несколько базовых элементов и основана на широком использовании графики. Графическое представление, иерархический подход и простота изучения позволили широко использовать эту методологию в целях анализа бизнес - процессов, на ранних стадиях разработки программного обеспечения для крупных проектов Европейского аэрокосмического агентства (Columbus), в качестве федерального стандарта США и как стандарт IEEE, FIPS, ISO.

Инструментарий **Design/IDEF**хорошо подходит для моделирования рабочих потоков и документооборота в страховых компаниях (модель процесса вычисления квот для различных категорий страховых полисов), в банках для моделирования обработки чеков, для изучения путей увеличения пропускной способности процесса обработки или для оптимизации работы парка машин, предназначенных для перевозки чеков из отделений банка в шифровальные центры для дальнейшей обработки. Уникальный проект использует эту методологию по управлению программой ядерных отходов. Агентство, отвечающее за размещение отработанного ядерного топлива и высокорадиоактивных ядерных отходов, используя этот проект, определяет возможности приемки, транспортировки и хранения ядерных отходов и начать подготовку для размещения отходов в геологическом репозитории в 2010 году.

#### Примеры использования надежных технологий управления

Информатизация налоговых служб

Современное налоговое законодательство содержит в себе огромный объем документов и законодательных актов, которыми необходимо владеть налоговому инспектору. Отчетная документация налогоплательщиков, представляемая ими по окончании отчетных периодов включает большое количество отчетных форм, которые необходимо проанализировать и проверить правильность исчисления налогов. Эта задача информатизации сложной системы с внешними связями и сложной внутренней структурой, которая должна функционировать в изменяющемся временном и законодательном пространстве и быть адаптированной ко всяким изменениям.

В основе решения данной задачи лежит объектно-ориентированный подход и комплексное рассмотрение всех проблем, позволяющий осуществить следующие функции:

* разработка компьютерной информационной технологии функционирования районных отделений налоговой инспекции;
* разработка комплекса анализа и прогнозирования экономического состояния региона;
* разработка интегрированной электронной базы данных налоговой полиции и налоговой инспекции;
* систематизация управления налоговой инспекции;
* пооперационный подход и внедрение четкого внутреннего разделения труда;
* калькуляция затрат на функционирование;
* определение методологии работы налогового инспектора;
* использование созданных технологий для процесса обучения сотрудников налоговой инспекции.

Технология использования электронных денег

Примером использования устойчивых и надежных информационных технологий в управлении может служить система VeriSmart, предоставляющая удобную и практичную систему использования смарт-карт. Это открытая система, являющаяся программно зависимой, работающая со многими клиентскими приложениями, от VeriFone Personal ATM до smart-phones и set-top-boxes, предоставляющая доступ к большому количеству различных приложений для работы со смарт-картами. Система не связана одной схемой работы со смарт-картами, единственным платежным приложением или стандартной конфигурацией оборудования. Модульная технология client-server системы VeriSmart облегчает установку системы и предоставляет свободу управления отличной программой смарт-карт, как для домашнего, так и для делового применения.

Согласно статистическим данным и прогнозам специалистов в 2000 году в применении находилось более $3.1 млрд. смарт-карт, а уже к 2005 году по всему миру ожидается увеличение объема транзакций по смарт-картам до $1.6 млрд. Такого рода рост показывает, что смарт-карты должны работать в двух направлениях. Во-первых, смарт-карты предоставляют пользователям высоконадежную, безопасную и практичную систему. Во-вторых, финансовые учреждения получают экономическую прибыль, экономят деньги, уменьшая необходимое количество наличности, получая большую защиту от краж.

Технология использования электронных денег VeriSmart предлагает новейшие приложения от проведения банковских операций в режиме “online” до совершения покупок в сети Internet и, при необходимости, возможность произвести upgrade.

Надежность использования технологии смарт-карт позволяет применить их в различных сферах деятельности. Во Франции банками уже выпущено более 20 млн. смарт-карт, которые применяются для дебитных и кредитных платежных операций, а также для оплаты разговоров в телефонах-автоматах. Более 80 млн. граждан Германии вскоре получат смарт-карты, содержащие информацию по медицинскому страхованию держателя. В Юго-Восточной Азии правительства Сингапура и Малайзии запускают в действие многоцелевую программу, основанную на технологии смарт-карт, объединяющую в себе схему электронного кошелька с программой скидок и привилегий, телефонную карточку и многое другое.

Открытая системаVeriSmart является программно зависимой и основана на использовании промышленно-стандартизованных протоколов. Она способна работать с большим числом приложений для клиентов и продавцов, предусмотрено объединение в компьютерную хост систему.

Ниже перечислены некоторые характеристики системы, которые позволяют быстро реагировать на изменения рынка смарт-карт:

* Открытая/Интероперативная Архитектура, работающая как с Windows NT, UNIX и другими платформами. Таким образом, VeriSmart может быстро реагировать на изменения рынка, быстро и легко добавляя или изменяя смарт-карт программы;
* Гибкость/независимость карт-схем: VeriSmart поддерживает многоцелевые приложения и способен виртуально работать с любыми карт-схемами; Mondex, VisaCash, Proton и другими. Это делает обслуживание более удобным;
* Надежность и совместимость работы приложений/решений: VeriSmart может не только поддерживать работу огромного спектра пользовательских приложений, но может также быть внедренной и абсолютно совместимой с backend host системами и сетями.

Поэтому VeriSmart представляет собой гибкое программное решение для управления смарт-карт программами. Надежность приложений обеспечивается простой в установке, легкостью и доступностью в применении, элементарностью усовершенствования.

Аналогичные системы, как правило, более сложные и дорогие. Ведь только upgrade такой системы может потребовать замены операционной системы, повторной установки всех приложений, а нередко это грозит и перевыпуском смарт-карт, переподготовкой и переориентацией баз, как производителей, так и клиентов.

Надежность применения VeriSmart позволяет объединить в себе усовершенствованные и надежные методы защиты для обеспечения полной безопасности проведения финансовых операций и передачи конфиденциальной информации. Для идентификации промышленных и персональных электронных ключейVeriSmart использует эллиптическо-инкриционный изменяющийся алгоритм. Таким образом, передача данных приложения между сервером VeriSmart и host компьютером надежно защищена.

Технология VeriSmart может быть интегрирована практически в любое электронное приложение и позволяет использовать смарт-карты с помощью смарт-карт - совместимых домашних персональных компьютеров, сетевых компьютеров, смарт-телефонов, пейджеров, set-top boxes, предназначенных для использования смарт-карт с использованием телевизора и др.

Примеры приложений, использующих методологию Workflow:

* Процедуры, связанные с кадровой работой;
* Документооборот;
* Требования по расходам;
* Покупки;
* Управление проектами;
* Информационный центр;
* Стандарты качества, например, ISO9000;
* Процедуры, связанные с обслуживанием клиентов.

Указанная методология Workflow может также использоваться и на правительственном уровне, а именно в виде приложений для центрального правительства:

* Служба занятости;
* Декларация о доходах;
* Рассмотрение апелляций;
* Обработка жалоб;
* Возмещение убытков.

Местные администрации могут использовать методологию Workflow для следующих целей:

* Выдача разрешений/лицензий;
* Продажа жилого фонда (право на покупку);
* Специальные требования;
* Ссуды на реконструкцию старых зданий;
* Контракты с поставщиками;
* Сбор задолженностей.

Ниже приведен список приложений, связанных с финансовой и юридической деятельностью.

* Финансовые услуги:
	+ Обработка требований по кредитам;
	+ Обработка страховых исков;
	+ Передача заложенного имущества;
	+ Пересылка пенсий.
* Профессиональный сервис:
	+ Аудиторское планирование;
	+ Апелляции, связанные, связанные с арендной платой и местными налогами;
	+ Передача собственности — нотариальное оформление актов о передаче имущества;
	+ Судебные процессы.

Надежные технологии системы обязательного медицинского страхования (ОМС)

Взаимозаменяемые (интероперабельные) карты на микропроцессоре Smart Card в силу их специфических характеристик гибкости и защищенности применяются в медицинских телекоммуникационных службах.

Использование Smart Card в ОМС

Можно выделить три главных направления телекоммуникации здравоохранения, входящих в стратегический план Информационной системы здравоохранения Европы:

* определение и выработка способов обмена и управления потоком информации, содержащейся в “мультимедийной карте” пациента;
* развитие телекоммуникационных предложений, которые увеличивают количество информации для лечащих врачей, для диагностики, лечения и управления службами, действующими как в интересах пациента, так и контролирующих органов;
* телемедицина для помощи пациентам, находящимся в условиях изоляции; выдача информации, касающейся профилактики и распознавания опасных заболеваний как гражданам, так и медицинским специалистам.

Межнациональное взаимное использование медицинских карт может быть достигнуто благодаря наличию общих характеристик и соответствия существующих стандартов, позволит улучшить работу всех участников обязательного медицинского страхования (граждан, медицинских работников, медицинской администрации и заинтересованных промышленных предприятий).

Применение в системе ОМС автоматизированной системы обработки информации с использованием "Евро-Рус-карт" предполагает привлечение и участие пациентов, страховых медицинских организаций системы ОМС, фондов ОМС, лечебно-профилактических учреждений (амбулаторных и стационарных), научно-исследовательских и учебных медицинских учреждений, аккредитованных медицинских ассоциаций, медицинских операторов (врачей и фармацевтов), органов управления здравоохранением, органов исполнительной власти.

Автоматизированная система обработки информации ОМС

Новая система медицинского обслуживания населения на основе автоматизированной системы обработки информации (АСОИ) базируется на следующих технологиях:

* Технология обеспечения полной и достоверной медицинской информации о пациенте, хранимой в базах данных лечебно-профилактических учреждений;
* Технология работы медицинских учреждений и их интеграция с другими социальными фондами и службами;
* Технология системы информатизации здравоохранения;
* Технология экономических методов управления медицинскими учреждениями;
* Технология телекоммуникационной связи между медицинскими операторами.

Необходимо рассматривать систему информатизации здравоохранения как часть единого информационного пространства и перепроектировать ее «снизу вверх» с непосредственным вовлечением медицинских учреждений, учитывая организационную, административную, имущественную, бухгалтерскую, управленческую и техническую автономию. Система контроля над управлением позволит осуществлять сбор данных о стоимости медицинских услуг и о деятельности медицинских учреждений.

Логическая модель АСОИ должна быть выполнена через стандартизацию и интеграцию способов получения аналитической информации, касающейся предоставления медицинских услуг, перестройку и нормализацию существующего информационного фонда (центральная, территориальная, районная и местная базы данных здравоохранения) и выдачу информации для облегчения количественного и качественного анализа расходов, явлений эпидемиологического характера, внедрение новых инструментов и методологий для активных действий органов ОМС и здравоохранения.

Своевременная и контролируемая информация о фактах, влияющих на стоимость системы здравоохранения, облегчает действия и повышает квалификацию механизмов контроля за расходами и упорядочению деятельности медицинских структур. Достижение этой цели осуществляется путем реализации оперативных проектов, касающихся центральных систем информатизации, таких как:

* аналитический учет всех оказанных услуг;
* автоматизации информации, касающейся движения средств;
* упорядочение балансов и смет медицинских учреждений;
* анализ основных данных показателей;
* сбора информации для принятия решений;
* определения бюджета расходов по услугам ирасходов в расчете на одного человека;
* нормализованная система управленческих отчетов.

Надежная технология автоматизированной системы обработки информации ОМС позволяет определить бюджет расходов на медицинские услуги в расчете на одного человека, инвестиционный бюджет, рациональное распределение финансовых ресурсов и обеспечит постоянный контроль за динамикой расходов на здравоохранение.

Данная система медицинского обслуживания, должна быть реализована при помощи:

* раздачи медицинских карт обслуживания граждан — "Евро-Рус-карт";
* создания телекоммуникационных связей между структурами и операторами, что обеспечивает циркуляцию медицинской информации и возможность получения сведений, касающихся важных фактов, имевших место в прошлом и находящихся в памяти местных сетей;
* создания телекоммуникационных связей между службами скорой и неотложной помощи;
* рационализация информатизации и доступ в Центры предварительной записи;
* интеграция с системами оплаты других услуг (коммунальные платежи и др.).

Распространение "Евро-Рус-карты" как средства идентификации и допуска к получению услуг в системе ОМС должно значительно упростить доступ гражданина к основным медицинским службам для получения наиболее эффективной медицинской помощи, а в перспективе — доступа к ряду других служб.

Использование в системе медицинского обслуживания "Евро-Рус-карт" позволяет:

* четко идентифицировать пациента;
* защитить личную информацию путем кодирования;
* оказать медицинскую консультацию;
* осуществить предварительную запись для получения определенных услуг;
* предоставить связь со службой скорой и неотложной помощи.

Система использования “Евро-рус-карт” позволяет там, где нет препятствий технологического, административного и правового порядка предоставить возможность доступа в систему оплаты (банковские расчеты), в систему страхования и службы местных организаций.

Эта карта является связующим звеном между населением, гражданами и оказанием медицинских услуг, обеспечивает им быстрый и уверенный доступ к информации и дает руководителям возможность эффективного и рационального контроля за реализацией функций, касающихся управления ресурсами и оказания медицинских услуг.

Получение в реальном времени медицинской информации (о предписаниях, предварительной записи, оказанных услугах) по телекоммуникационной сети АСОИ с применением медицинской карты и активизация систем мониторинга, позволяют осуществлять эффективный контроль качества медицинских услуг.

Рационализация существующей информационной и технологической инфраструктур должна быть направлена на:

* создание нового центрального нормализованного банка данных здравоохранения;
* доведение этих данных до региональных периферийных структур нижнего уровня (медицинские учреждения);
* организацию сбора данных, касающихся оказания амбулаторных, диагностических, специализированных, фармацевтических, больничных и др. услуг;
* облегчение оформления документации, требуемой медицинскими органами (полугодовые отчеты, межтерриториальные компенсации и т.д.);
* развитие проектов по медицинским картам, системам взаимосвязи, многофункциональности периферийных структур;
* создание прочной основы для комплексной реализации системы информатизации здравоохранения.

Архитектурная модель АСОИ позволяет организациям, вовлеченным в новый процесс информационной автоматизации в здравоохранении, сотрудничать между собой и с органами ОМС и органами здравоохранения.

Таким образом, телекоммуникационная сеть между компьютерами, установленными в органах обязательного медицинского страхования и медицинских учреждениях, соединенная с "Евро-Рус-картами", образуют комплекс информационных технологий управления, на которых базируется вся технологическая перестройка системы информатизации здравоохранения.

При использовании существующих и перспективных телекоммуникационных сетей при создании системы необходимо уделить особое внимание безопасности и защищенности передачи данных, применяя те же критерии, что и в информационной сети, используя при этом возможности карты на микропроцессоре.

Автоматизация потоков информации, поступающая от медицинских учреждений при помощи телекоммуникаций, явится первым шагом к упрощению управления, даст возможность осуществить более действенный контроль и облегчить взаимосвязь со всеми заинтересованными организациями. Все данные будут автоматически заноситься всоздаваемый архив данных с тем, чтобы можно было получить ранее сделанные анализы и другую информацию, нужную органам обязательного медицинского страхования.

АСОИ улучшает качество предоставляемых учреждениями здравоохранения услуг населению; повышает обеспеченность процесса лечения современными лекарственными средствами и лечебными процедурами, в перспективе изделиями медицинского назначения; повышает уровень обеспеченности лечебных и профилактических учреждений диагностической и медицинской аппаратурой; обеспечивает полную и объективную информацию о качестве и полноте диагностики, профилактики и лечении заболеваний, профессиональном уровне медицинского персонала, о соответствии уровня подготовки и переподготовки кадров в соответствии с современными требованиями, формирует базы для лечебно-профилактических и фармацевтических учреждений.

Надежная реализация требований нормативных баз обеспечивается архитектурой АСОИ, структурой взаимодействия в системе ОМС и выбором вычислительных средств и протоколами взаимодействия и доступа к информации, а также требованиями к защите информации.

Рассмотренные методы структурирования процессов (IDEF) и обеспечения надежных потоков (Workflow) в комплексе с рассмотренными ранее методами достижения надежности с использованием базы интеграции процессов в информационных технологиях управления создают предпосылки для комплексного понимания сущности надежности АС, гарантирующих заданные свойства процесса менеджмента.

Однако процессы развития элементной и системной надежности существенно влияют на изменения методов достижения надежности и должны быть постоянно в поле зрения менеджера и взаимодействующих с ним администраторов АСУ.

Для этой цели необходимо осуществлять систематический контроль за такими системами.

**Программное обеспечение как надежная система технологий управления**

Назначение любой вычислительной системы - решать задачи. Для каждой задачи это означает - выполнить работу (в частности, вычислить значения некоторых величин), определяемую программой решения этой задачи при условии, что входные величины имеют заданные значения.

Сам вычислительный процесс представляет собой сложную систему управления с многочисленными контурами управления. Рассмотрим контур, который представляет собой программу. В каждой программе можно выделить конструктивные средства, выполняющие основное назначение программы, и управляющие средства - средства, обеспечивающие эффективное исполнение конструктивных средств.

В реальных системах часть управляющих функций выполняет операционная система. Тем не менее, и в программе всегда есть управляющие средства. Примером таких средств являются операторы переходов, операторы цикла и т.п. Конечно, целью программы является получение результата решения задачи, но в реальных условиях достоверность полученного результата (даже, если он получен) может быть разной, да и требования, которые предъявит программа к ресурсам (время, стоимость и т.д.) в процессе выполнения могут оказаться чрезмерными,

Поэтому, рассматривая программу как систему управления, можно сформулировать такие цели управления:

* повышение достоверности результата при ограничениях на время выполнения программы, на стоимость решения, на конфигурацию внешней памяти и т.п.;
* снижение времени решения при приемлемом уровне достоверности ;
* снижение стоимости решения задачи при заданных границах времени выполнения программы, достоверности результата и т.д.

Такая система управления схематично показана на рис. 4.11.

R

**S**

# A

# M

**C1**

**C2**

N

Y

X

Рис. 4.11. Модель программы

|  |  |
| --- | --- |
| N - внешние воздействияX - входные величиныY - результатR - конструктивные средства | S - управляющие средстваM - модель результатаA - блок выбора реакцииC1, C2 - управляющие воздействия |

Рассмотрим внешние воздействия N, показанные на схеме. Они могут быть запланированными или неожиданными.

Внешние воздействия могут быть разного происхождения:

* от технических средств;
* от человека (оператора);
* из других программ (подпрограмм);
* вводимая в динамике выполнения информации.

Внешнее воздействие будем называть корректным, если значение его принадлежит ареалу - заранее оговоренному множеству допустимых значений - и поступает оно в программу в оговоренное время. В противном случае воздействие будем называть некорректным. Аналогично, и значения входных величин могут быть корректными и некорректными.

Понятно, что проникновение в программу некорректных значений, будь то внешнее воздействие или входная величина, чаще всего приводит к неправильному функционированию программы, к неправильному результату. Поскольку программа обычно входит в состав комплекса программ, то полученный некорректный результат становится некорректным входным значением и т.д., что может вызвать цепную реакцию неверных результатов и привести к тому, что действия системы станут непредсказуемыми. Можно привести много реальных примеров такого типа; неудача при первом запуске американского исследовательского корабля на Венеру, когда пришлось взорвать этот корабль из-за серьезного отклонения от курса; смертные случаи из-за ошибок в медицинском программном обеспечении; авиакатастрофы из-за ошибки одной из программ комплекса проектирования самолета; уничтожение 72 французских шаров-зондов с измерительными приборами (из 115 запущенных) из-за того, что была послана не та команда управления и т.п. Перечень таких примеров может быть продолжен, хотя не всегда, конечно, ошибки приводят к таким серьезным последствиям.

Застраховаться от таких ситуаций можно только повышением надежности всего вычислительного процесса и, в частности, программного обеспечения.

Позже мы рассмотрим методы повышения надежности программ и покажем целесообразность введения дополнительных контуров управления для достижения этой цели.

Заметим, что для рассмотренной схемы управления можно пренебречь разницей между входными величинами и внешними воздействиями, так как и те, и другие представляются для программы некоторой информацией, подлежащей анализу и/или обработке (случай, когда внешние воздействия искажают саму программу, здесь не рассматриваем). Поэтому, для удобства изложения мы иногда будем говорить о входных значениях, имея в виду и те, и другие.

Рассмотрим управляющие средства контура. Они осуществляют управление программой, олицетворяя и обратные связи в программе.

М - модель результата. Здесь проверяется, соответствует ли результат работы средств R спланированной ранее модели, например, удовлетворяет ли он условию 0 ≤ Y ≤ 1.

Для того, чтобы модель могла функционировать, в программе должны быть средства измерения результата, обеспечивающие обратную связь, и средства сравнения.

А - блок выбора реакции. В зависимости от результата, выдаваемого моделью, выбирается тот или иной метод воздействия на ход выполнения программы.

Реализация выбранного метода управляющего воздействия показана блоками С1, С2. Таких блоков в сложных случаях может быть больше, но в каждый конкретный момент может выполняться только один из них.

Управляющее воздействие может заключаться в передаче управления тому или иному участку программы (иначе говоря, к изменению значений счетчика адреса команд), в изменении значения какой-то переменной величины (чаще всего - логической), в формировании той или иной конструкции. В отдельных случаях это может быть и сообщение оператору.

Простейшим примером управляющего средства может служить оператор 1F. Например, 1F (Х=1) ТНЕN GО ТО А, ЕLSЕ В:=1

Здесь роль блока М играет вычисление предиката (Х=1), блок А определяет, какой из путей THEN или ЕLSЕ выбрать, а управляющие воздействия: GO TO A и B:=1. Как уже говорилось, целью программы, как системы управления может быть повышение достоверности результата или, по крайней мере, достижение приемлемого уровня достоверности. Для этого программа должна иметь достаточно мощные управляющие средства, обеспечивающие кроме измерения нужных характеристик, выбора и реализации управляющих воздействий, еще и повышение надежности программ.

Само выражение "повышение надежности" предполагает знание того, что такое надежность и как ее измерить (чтобы убедиться, что надежность повысилась). К сожалению, единой точки зрения на надежность программы и способы ее измерения пока нет.

**Надежность программ. Разные подходы**

Вопрос определения и повышения надежности программ приобретает в последнее время особую важность в связи с возросшим уровнем сложности применения ЭВМ.

В подходе к проблеме надежности программ иногда ориентируются на применение основных положений теории надежности аппаратуры к программам. Вводится понятие отказа программы - это ошибка в программе, вызывающая неправильное функционирование программы в каких-либо условиях. Соответственно может рассматриваться наработка на отказ, время восстановления т.д.

Надо, однако, представлять себе разницу между аппаратурой и программой, которая заключается, прежде всего, в том, что аппаратура отказывает с определенной вероятностью, даже если она не функционирует; программа же не изнашивается, поломка программы невозможна. Практически не влияют на функционирование программ и производственные дефекты (например, ошибки копирования системы во время ее переноса), так как они редки, быстро обнаруживаются и легко устраняются. Ошибки в программах проявляются как систематические и далеко не случайные. Кроме того, в разных применениях одна и та же программа может демонстрировать различную интенсивность отказов, что вызывает серьезные трудности при попытке использовать методы теории надежности аппаратуры для анализа надежности программного обеспечения.

Существует и другой подход, ориентированный на анализ структуры программы с учетом специфики ее применения. Важную роль при таком подходе играет управляющий граф программы. Управляющий граф (граф переходов) программы определяется как ориентированный граф, вершины которого соответствуют одному или нескольким операторам этой программы, а дуги отражают возможности передач управления между соответствующими операторами.

Управляющий граф программы является моделью этой программы, отражающей логику выполнения этой программы, сложность ее функционирования при любых значениях исходных данных. Изучение управляющего графа позволяет понять закономерности поведения программы в динамике, позволяет обнаружить многие некорректности в программе, позволяет преобразовать программу с целью ее оптимизации. Управляющий граф является основой для определения стратегии тестирования программы, ее анализа, отладки.

В литературе можно встретить различные определения надежности программ. Одну группу определений можно свести к такому определению: надежность программы есть качество ее отладки или число оставшихся в программе ошибок.

Вторую группу определений подытоживает определение, приводимое Г.Майерсом: Надежность программного обеспечения есть вероятность его работы без отказов в течение определенного периода времени, рассчитанная с учетом стоимости (важности) для пользователя каждого отказа.

Наличие таких различных подходов, как к самой проблеме надежности программ, так и к определению надежности, неизбежно демонстрирует, что это - понятие не простое, включающее различные аспекты.

Надежная программа должна обладать двумя основными свойствами:

1. Программа должна правильно функционировать при корректной внешней среде. Мы подразумеваем, что задание на проектирование программы оговаривает, что такое правильное функционирование и для какой внешней среды (это и значения входных величин, и параметры управляющих программ, операционной системы, виртуальной и реальной машины, и внешние воздействия и т.п.) должно быть обеспечено это правильное функционирование. Естественно, оно должно быть обеспечено и для конструктивных средств программы и для управляющих средств.

Такое свойство программы назовем безошибочностью.

Естественно, добиться абсолютной безошибочности программы не представляется возможным. Поэтому по отношение к такому свойству, как безошибочность, можно говорить о количестве оставшихся в программе ошибок (еще не обнаруженных); заметим, что существуют методы, позволяющие оценить это количество. Соответственно, можно говорить об уровне безошибочности.

2. Программа должна однозначно реагировать на некорректную внешнюю среду. Для каких-то случаев она должна продолжать функционирование (отметив, что столкнулась с некорректностью), для каких-то должны быть выбраны обходные пути, для каких-то может быть выбран путь снижения производительности (деградации). В каких-то случаях программа может прекратить выполнение, но в любом случае не должно быть непредсказуемого поведения, все отклонения от нормы должны быть заранее предусмотрена, недопустимо распространение некорректностей в другие программы.

Такое свойство программы назовем помехоустойчивостью.

Таким образом, можно говорить о двух видах причин, снижающих надежность функционирования программы.

Во-первых, внутренние причины, ошибки в конструктивных средствах программы, приводящие к нарушению правильности, к тому, что при некоторых корректных значениях входных величин (включая и внешние воздействия) программа либо выдает неправильный результат, либо вообще не завершается. Иначе говоря, снижается уровень безошибочности программ.

Во-вторых, внешние причины, недостаточная защита от некорректных значений входных величин и внешних воздействий. Строго говоря, это можно считать ошибками в управляющих средствах программы, снижающими помехоустойчивость программы, достоверность получения правильного результата.

В соответствии с этим, повышение надежности программы связано либо с повышением уровня безошибочности программы, либо с повышением уровня помехоустойчивости, либо и с тем, и с другим. В дальнейшем мы рассмотрим методы повышения надежности именно с этих позиций.

#### Технологии повышения безошибочности программ

Методы повышения безошибочности программ можно разделить на две группы: методы, применяемые при проектировании и кодировании программ, и методы, применяемые при отладке готовой программы.

К первой группе методов относятся, в частности:

* использование принципов модульного программирования;
* использование принципов структурного программирования;
* применение языков высокого уровня;
* повышение уровня непроцедурности программы;
* методы синтеза программ по спецификациям и методы автоматического доказательства программ.

Цель всех этих методов - предупредить ошибки, не допустить появления ошибок в готовой программе. Совершенно ясно, что предупреждение ошибок самым серьезным образом повышает вероятность правильности программ. Специалисты называют разные цифры стоимости нахождения, и исправления ошибок в программе в зависимости от времени их обнаружения. Обычно утверждается, что раннее обнаружение дешевле в 10 и более раз.

Но какими бы методами проектирования ни пользовался программист, невозможно гарантировать безошибочность спроектированной только что программы. Программа должна обязательно пройти и этап отладки, в результате которого вероятность правильного функционирования программы окажется выше. Ко второй группе методов повышения безошибочности программ относятся разнообразные методы отладки.

Здесь мы не рассматриваем организационные методы руководства разработкой программного обеспечения, хотя их влияние на надежность создаваемых программ может иногда оказаться решающим.

#### Технологии, применяемые при подготовке программы

* Модульное программирование предполагает, что программа разбивается на отдельные части-модули, причем разбиение должно в той или иной мере удовлетворять требованиям, среди которых отметим:
* ограничение размера модуля (обычно, в пределах 20-100 операторов);
* семантическое выделение в отдельные модули (неформальная функционально-структурная декомпозиция);
* относительная независимость модуля. Связи между модулями как по управлению, так и по информации должны быть минимальными.

Однако нельзя слишком строго следовать принципам модульности, иначе это может привести к слишком большим накладным расходам. Важнейшую роль в модульном программировании играют интуиция и опыт программиста.

Хорошая модульная программа имеет высокий уровень безошибочности потому, что связи между модулями достаточно просты и могут быть прослежены за столом. В то же время исправление ошибки в одном из модулей не должно по возможности приводить к корректировке других модулей.

Очень важно уметь использовать в программе стандартные модули (подпрограммы). Эти модули заранее отлажены, опробованы в разных применениях и, следовательно, содержат меньше ошибок, чем написанные специально для этой программы.

Можно сказать, что модульное программирование понижает сложность разрабатываемой программы, представляя ее в виде совокупности менее сложных модулей с несложными связями между ними. Тем самым упрощается анализ программы, поиск ошибок в ней и модификация.

Обычно программа разделяется "географически", т.е. модуль создается из операторов, компактно расположенных в программе. Однако, если привлечь технику получения остаточных программ и методы вертикального слоения, можно создавать модули и по другим (более функциональным) принципам.

Разделение на модули предполагает максимальную независимость модулей путем усиления внутренних связей в каждом модуле (повышения прочности модулей) и ослабления связей между модулями (и по данным и по управлению), применяя формализованный механизм передачи параметров. Вот какую цель формулирует академик А.П.Ершов: "Нам надо научиться очень тщательно и доказательно писать небольшие программы и в то же время уметь быстро, уверенно и безошибочно монтировать огромное комплексы программ из полуфабрикатов”.

Структурное программирование - это технология программирования, в которой разрешено использование только трех видов конструкций: последовательность, выбор, итерация.

Конструкции в программе могут комбинироваться друг с другом так, как этого требует программа. Всякая программ, построенная в терминах этих конструкций, может быть постепенно преобразована в одну из этих конструкций. Так реализуется идея вложенной структуры, являющаяся важной в структурном программировании.

Программа, построенная таким образом, является модульной в строгом смысле слова. Потому что, основное в модульном программировании - это возможность замены отдельного модуля его эквивалентом без последствий в остальной части программы.

Принципы структурного программирования реализованы во многих языках программирования. При программировании должны использоваться либо обычные конструктивные операторы для выполнения в линейном порядке, либо перечисленные ранее управляющие средства выбора и повторения.

Структурные программы обладают многими свойствами, облегчающими поиск ошибок. Тем самым повышается надежность программ. Значение структурного программирования не исчерпывается сказанным. По сути, это - дисциплина программирования, представляющая собой, постепенное превращение программирования из ремесла в науку. Наиболее очевидная выгода - рост производительности и уменьшение процента ошибок.

Языки высокого уровня позволяют готовить программу в терминах более соответствующих самой задаче. При программировании на языке высокого уровня не надо учитывать специфику реализации, не надо (или почти не надо) заботиться о распределении ресурсов. В процессе программирования усилия сосредотачиваются на анализе конструктивных средств программы, на связях между ними, на формировании более сложных конструкций. Разные языки высокого уровня предоставляют разные возможности по созданию правильных программ, но в любом случае эти возможности выше, чем в языках типа Ассемблера.

Основные параметры, по которым можно сравнивать языки высокого уровня, с целью выбрать "более надежный" - читабельность подготовленных программ и отсутствие особенностей языка, провоцирующих ошибки.

Непроцедурность программ позволяет уйти от подобного (а значит, чреватого ошибками) расписывания последовательности действий для выполнения того или иного фрагмента программы. Непроцедурное программирование фрагмента означает задание спецификаций входа и выхода этого фрагмента без указания алгоритма преобразования. Иначе говоря, программист оговаривает, ЧТО НАДО СДЕЛАТЬ, не указывая КАК НАДО СДЕЛАТЬ. Реализация фрагмента по его спецификациям выполняется автоматически. Сама программа при этом не меняется. Примером метода повышения непроцедурности может служить использование макросредств - механизма получения открытых подпрограмм, настроенных на значения параметров. Другой пример - использование объектно-ориентированных языков. Предварительная и независимая отлаженность стандартных алгоритмов, привлекаемых для расшифровки спецификаций, позволяет повысить безошибочность подготовленной программы.

Методы синтеза программ и автоматического доказательства их правильности ориентируются на формальные методы, которые позволили бы получать безошибочные, абсолютно правильные программы. По-видимому, построить практическую технологию создания произвольных абсолютно правильных программ невозможно. Но для некоторых классов программ такие исследования ведутся. Тем не менее, в ближайшее время нельзя ожидать реальных синтезаторов и верификаторов для сколько-нибудь практического применения.

**Технологии отладка программ**

Одна из аксиом программирования гласит: Любая нетривиальная программа содержит хотя бы одну ошибку. Практика показывает справедливость этой аксиомы. Как бы хорошо ни была подготовлена программа, какие бы методы ни применялись при построении программы, следует исходить из "презумпции виновности".

Для обнаружения и устранения ошибок выделяется целый этап - отладка программ. Большинство специалистов считают, что отладка (вместе с тестированием) занимает 50% времени разработки программ.

Цель отладки - изменить программу таким образом, чтобы она была в состоянии решить запланированную задачу.

В процессе отладки можно выделить следующие этапы: тестирование, анализ, внесение изменений. Эти этапы повторяются многократно до тех пор, пока не будет принято решение о прекращении отладки и передаче программы в эксплуатацию.

Цель тестирования - продемонстрировать, что ошибка в программе есть, так что, если результат прогона какого-либо теста совпадает с предполагаемым результатом, считается, что цель тестирования не достигнута. При тестирований может выполняться не вся программа, а только ее часть - от одной заданной точки программы до другой. Существенным для тестирования является построение тестовых наборов, то есть тех значений, которые являются кодами для тестирования.

Само тестирование разделяют на три фазы: составление (генерация) тестовых наборов, выполнение программы на этих наборах, оценка результатов выполнения программы. Для простых случаев все три фазы выполняются "вручную", без машины. Но серьезное тестирование должно, конечно, включать в себя работу на машине. К сожалению, в общем случае полную систему тестовых наборов построить автоматически невозможно.

Хотя цель тестирования - констатация наличия ошибки, тем не менее, цель выполнения каждого отдельного тестового задания - получить корректный результат. Для этого, если результат оказался некорректным, а это определяется анализом результата - следующим этапом отладки, то выполняются какие-либо изменения, направленные на устранение обнаруженной некорректности.

Представим отладку программы в виде контура управления, в котором объектом управления является отлаживаемая программа Р. Целью такой системы является повышение уровня безошибочности программы.

На схеме, представленной на рис. 4.12, субъект управления представляет собой отладочные средства, включают кроме программ еще и человека, выполняющего многие неформализованные действия.

**P**

**D**

# С

# А

# М

Y

T

Рис. 4.12. Схема отладки

Т- это задание на тестирование, включающее, естественно, и тестовый набор. Блок М определяет корректность выполнения тестового задания, блок А определяет причину некорректности, блок С осуществляет целенаправленное изменение. Изменение может быть направлено на программу Р, (с тем, чтобы добиться корректного выполнения тестового задания или включить в программу инструментальные вставки - счетчики, команды печати, команды прерываний и т.п. - для облегчения последующего анализа) или на само тестовое задание (если ошибка обнаружена в задании), или на порядок выполнения тестовых заданий (если задание уже выполнилось корректно или полученных результатов недостаточно для определения причины некорректности).

Таким образом, блоки М и А олицетворяют анализ, а блок С - внесение изменений.

Несмотря на трудности, существующие при генерации тестовых наборов и при решении связанных с этим оптимизационных задач, самым неформализуемым, самым сложным, требующим самых больших интеллектуальных усилий этапом является анализ, анализ ситуации, возникающей в результате выполнения (не обязательно некорректного) тестового задания.

В традиционных методах отладки объектом управления является сама программа Р или ее инструментированная модификация. С появлением аппарата смешанных вычислений можно рассматривать и такие методы отладки, где объект управления - остаточная программа.

Остаточная программа - понятие, введенное А.П. Ершовым в связи с определением смешанных вычислений. Остаточная программа - это программа, получаемая в результате адаптации исходной программы к некоторым дополнительным условиям. Обычно эти условия представляют собой задание значений некоторых из входных величин или ограничения, дополнительно накладываемые на ареалы этих величин (сужение ареалов). Уровень адаптации зависит от самой исходной программы, и от возможностей смешанного вычислителя. Разные смешанные вычислители по одинаковой дополнительной информации могут породить, вообще говоря, разные остаточные программы.

Не всегда дополнительные условия определяют требования к значениям входных величин. Условия могут быть достаточно сложными и касаться разных объектов, свойств, или функций программы. Важным частным случаем представляется, например, такое условие - программа при выполнении должна проходить через заданный оператор. Адаптация к такому условию реализуется преобразованием к новой программе, из которой, по сравнению с исходной, удалены пути, не проходящие через оператор. Назовем такую новую программу - остатком исходной программы по конкретному заданному оператору.

Использование остаточных программ и остатков по заданным операторам может упростить анализ и уменьшить объем тестирования. Методика отладки с помощью остаточных программ и остатков по заданным операторам опирается на предположение, что более простой объект легче анализировать.

Пусть, например, для устранения причины ошибки, обнаруженной при тестировании и диагностировании при анализе, предполагается откорректировать некоторый оператор программы. Построим остаток программы по этому оператору - получится более простая программа, анализ которой потребует меньше усилий. Отброшенные участки программы не зависят от предполагаемой корректировки и не влияют на состояние изменяемого оператора. Можно показать и другие достоинства применения остаточных программ и остатков для отладки. В частности, можно уменьшить объем тестирования.

#### Технологии повышения помехоустойчивости программ

Помехой будем называть некорректное значение внешнего воздействия или входной величины. Ранее мы уже определили, что такое корректное внешнее воздействие: его значение должно принадлежать ареалу и поступать оно в программу должно в оговоренное время (можно, по-видимому, рассматривать внешнее воздействие как пару - значение и время появления; тогда и ареал должен состоять из двух компонентов, один из которых оговаривает допустимые значения, размеры воздействия, а другой - допустимые моменты времени).

Уровень помехоустойчивости определяется способностью программы не допустить влияния помех на конечный результат решения задачи. Конечно, более высокий уровень помехоустойчивости требует больших затрат времени, объема памяти и т.п.

Помехоустойчивость программы может быть повышена при использовании различных видов резервирования, т.е. введением избыточности.

Избыточность вводится для обнаружения, изоляции и устранения помех. Избыточность, как и резервирование, может быть временной, информационной, программной (алгоритмической), эргатической.

Временная избыточность предполагает планирование функционирования программы, при котором создается резерв времени для выполнения заданных функций. Этот резерв позволяет в случае необходимости повторять выполнение тех или иных фрагментов программы, включать дополнительный контроль, улучшить сервис при эксплуатации программ.

Информационная избыточность реализуется введением дополнительной информации, сопровождающей основную при передачах, при обработке, при отображении. Это могут бить контрольные суммы, корректирующие кода и т.д. Цель введения избыточной информации - обнаружение помех, а если удастся, то и устранение обнаруженных ошибок.

Программная избыточность является средством для защиты от помех, обнаружения и устранения ошибок. Средство это реализуется программным путем, введением дополнительных алгоритмов контроля в программе решения задачи.

Эргатическая избыточность предполагает выдачу сообщений, которые позволяют человеку повлиять на ход решения задачи с целью устранения всевозможных помех.

Наиболее распространенные методы повышения помехоустойчивости программ используют различные виды избыточности:

* временную, поскольку для реализаций методов требуются дополнительное машинное время;
* информационную, поскольку для проведения контроля необходима дополнительная информация;
* программную, поскольку реализация методов программная;
* эргатическую, поскольку не всегда удается устранить помеху автоматически; в этих случаях может помочь только вмешательство человека.

Помехоустойчивость программы можно повысить применяя способы повторения выполнения фрагментов программ и организации защиты от проникновения помех в программу.

#### Повторение выполнения

Чаще всего применяются программистами такие два способа повторения выполнения фрагментов - двойной счет и контрольные точки.

Двойной счет организуется обычно по такой схеме. Отдельный фрагмент задачи решается дважды, затем результаты сравниваются, если они совпадают, то решение считается правильным, и переходят к следующему фрагменту программы. Если же результаты не совпадают, то может быть выполнена третья попытка, и если результаты третьей попытки совпадают с одним из ранее полученных результатов, то они считаются правильными.

Здесь мы опять встречаемся с контуром управления. Объектом управления в таком контуре является фрагмент программы. Цель контура - достижение высокой достоверности решения ценой потери времени.

Существуют модификации метода двойного счета, например, второй просчет может бить выполнен по другому алгоритму; делается это в предположении, что помеха может выборочно влиять на конкретный алгоритм, на конкретную операцию.

Контрольные точки организуются в программах, время выполнения которых достаточно велико - несколько десятков минут, например. Цель этого способа - экономия времени выполнения программы при достаточно высоком уровне достоверности полученного результата.

Сам способ заключается в том, что в программе расставляются так называемые контрольные точки, в которые помещается обращение к процедуре "фотографирования" оперативной памяти во внешнюю память. Контрольные точки расставляются таким образом, чтобы время выполнения программы между контрольными точками было в пределах 5-10 минут. Если в какой-то момент в промежутке между контрольными точками произошел сбой оборудования или появилась какая-либо помеха, не позволяющая ожидать достоверный результат решения, то программу можно запустить не с самого начала, потеряв все время решения, а с предыдущей контрольной точки, предварительно восстановив соответствующее состояние оперативной памяти с магнитной ленты. Потеряно будет практически только время, прошедшее после организации соответствующей контрольной точки.

Конечно, организация контрольных точек далеко не всегда такое простое дело, как могло бы показаться. Существует много проблем, которые приходится при этом решать, но чем-то же надо платить за экономию времени решения задачи при неустойчиво работающей аппаратуре.

#### Технологии защиты от проникновения помехи

Как уже говорилось, проникновение некорректных значений (помех) в программу чревато очень серьезными последствиями. Поэтому обычно предпринимаются те или иные условия по организации контроля. Программисты выполняют эту работу каждый по-своему. В результате оказывается, что даже для одинаковых входных величин средства защиты от помех, построенные разными программистами отличаются как затратами на создание этих средств, так и тщательностью защиты.

К сожалению, первопричиной плохого контроля в программах, является недостаточное представление о последствиях этого плохого контроля, о реальности помех, излишней самоуверенности с одной стороны и удивительной доверчивости с другой.

Программирование с защитой от помех обычно отнимает слишком много времени при подготовке программы (по некоторым оценкам, больше 20%).

Затраты эти можно существенно снизить, если предложить программисту автоматизированный способ подготовки средств контроля.

Ниже мы рассмотрим фильтрацию - автоматизированный процесс, предотвращающий проникновение помех в программу. Выделяя этот процесс из других контролирующих процессов, мы исходим из тезиса необходимости как можно более раннего обнаружения ошибки. Кроме того, мы считаем, что контроль должен быть управляемым со стороны программиста, иначе говоря, программист сам должен определять, что надо контролировать и когда надо контролировать. Мы можем только выдать соответствующие рекомендации.

#### Технологии фильтрации

Назовем фильтрацией величины Х из программы Р автоматизированный процесс, предотвращающий проникновение некорректных значений величины Х в программу Р. Этот процесс включает проверку корректность значения величины Х и реакции на некорректность такого значения. Программные средства, реализующие процесс фильтрации назовем фильтрами. Существенным для фильтра в применении к практическому программированию является его ориентация на спецификацию контролируемой величины, включающую в себя ареал величины и описание реакций на некорректность значения этой величины.

Ареал. Любая программа ориентирована на явно или неявно заданное множество значений своих входных величин (область определения, множество допустимых значений). Когда программа используется совместно с другими программами (в системе, в комплексе), это множество значений может оказаться более конкретным, более узким.

Ареалом величины Х из программы Р называется множество значений, которые могут быть присвоены Х, исходя из назначения и условия применения программы Р.

Определение подчеркивает, что разные условия применения программы могут определять разные ареалы величины.

Понятие ареала распространяется не только на входные величины, но и на переменные, и на результирующие. Заметим, что для одной и той же величины Х в разных местах программы могут быть заданы разные ареалы.

Реакция на некорректность контролируемой величины может быть разной:

* сообщение об ошибке;
* замена некорректного значения стандартным значением;
* замена ближайшим (по некоторому критерию) значением;
* обращение к подпрограмме пользователя;
* прекращение вычислений и т.д.

В любом случае реакция на некорректность должна быть предусмотрена заранее при составлении спецификации величины.

Фильтр - стандартизованное средство, автоматически настраиваемое на спецификацию (точнее, на ареал) фильтруемой величины. Настройка эта может выполняться как при использовании фильтра, так и при его построении. Для того, чтобы выполнить фильтрацию, необходимо узнать:

* имя фильтруемого объекта или значение объекта;
* спецификацию объекта;
* точку пуска фильтрации.

Реализация фильтра зависит от того, на каком этапе определяется фактическое значение фильтруемой величины. Так, значения параметров макрокоманды определяются на этапе подготовки программы. Могут быть заранее известны и некоторые параметры подпрограмм, процедур. Фильтры, реализующие фильтрацию таких величин, могут быть включены в процессоры подготовки программ - в макрогенераторы, в сборщики (редакторы связей), в компиляторы, в смешанные вычислители. Такие фильтры представляют собой универсальные процедуры (подпрограммы, макроопределения), входной информацией для которых являются контролируемое значение, ареал величин, описание реакции на некорректность значения. Результатом фильтрации является только ответ, корректно ли значение, а в случае некорректности - оговоренная заранее реакция. Время фильтрации на этом этапе не является критическим, если же значение величины при подготовке программ неизвестно, определится оно при выполнении программы, то и фильтрация должна выполняться во время выполнения программы. Появляются новые контуры управления, цель которых - фильтрация переменных величин.

Фильтры, проверяющие переменные величины, включаются в саму программу процессорами подготовки программ: трансляторами, макрогенераторами. Здесь могут быть два варианта:

1) Процессор подготовки программ настраивает универсальную заготовку на ареал фильтруемой величины, и полученный в результате настройки фильтр включается в нужное место строящейся программы. Включаются и сам фильтр, и спецификация объекта, и имя объекта.

2) Процессор подготовки программ включает в программу обращение к стандартному, не настраиваемому фильтру. Это обращение сопровождается ссылкой на спецификацию фильтруемой величины и должно включаться в нужном месте.

Постоянное возрастание объемов и сложности производимого программного обеспечения выдвигает проблему повышения надежности программного обеспечения в ряд основных проблем разработки программных средств.

К сожалению, сегодня не существует не только практических методов оценки надежности программных изделий, но и четкого определения надежности программ. Многочисленные методы, о которых можно интуитивно говорить, как о повышающих надежность программ, не объединены методически и не поддаются сравнению с точки зрения их эффективности.

1. Если для АСУ сформулирована составная функция наиболее общего вида, то соответствующая ей функциональная подсистема совпадает с системой в целом. [↑](#footnote-ref-1)
2. **Энтропия** (от греч. entropía - поворот, превращение), понятие, впервые введенное в термодинамике для определения меры необратимого рассеяния энергии. Энтропия широко применяется и в других областях науки: в статистической физике как мера вероятности осуществления какого-либо макроскопического состояния; в ***теории информации*** как мера неопределенности какого-либо опыта (испытания), который может иметь разные исходы. Эти трактовки энтропии имеют глубокую внутреннюю связь. Например, на основе представлений об информационной энтропии можно вывести все важнейшие положения статистической физики.

В термодинамике понятие "энтропия" было введено Р. Клаузиусом (1865), который показал, что процесс превращения теплоты в работу следует общей физической закономерности - второму началу термодинамики. Его можно сформулировать строго математически, если ввести особую функцию состояния - энтропию. [↑](#footnote-ref-2)
3. Парадигма (от гр. пример, образец) — теория (или модель, тип постановки проблемы), принятая в качестве образца решения исследовательских задач [↑](#footnote-ref-3)
4. **Декомпозиция** - расчленение исходной системы на относительно обособленные части [↑](#footnote-ref-4)