



ОБЪЕДИНЕННЫЙ  
ИНСТИТУТ  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

Дубна

00-187

P10-2000-187

В.М.Добрянский, В.Н.Самойлов, А.В.Чекер

ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ И ТЕХНОЛОГИЯ  
СОЗДАНИЯ АВТОМАТИЗИРОВАННОЙ  
ИНФОРМАЦИОННОЙ СИСТЕМЫ  
ДЛЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ ПРОЦЕССОВ

Направлено в журнал «Автоматизация проектирования»

2000

## Введение

Разработка информационных систем становится широко распространенной задачей, решаемой в самых различных сферах человеческой деятельности. Определены типовые схемы информационных систем, их наиболее характерные области применения. Однако в условиях быстрого развития вычислительной техники и прикладного программного обеспечения, усложнения научных экспериментов и промышленных процессов потребность в совершенствовании информационных систем резко возрастает.

В работах [1, 2, 5] предложены новые подходы к построению систем моделирования сложных физических процессов и сформулированы требования к автоматизированной информационной системе (АИС), интегрированной в этот процесс. В них автоматизированная информационная система моделирования сложных процессов рассматривается не только как инструмент решения задачи, но и как автоматизированное средство постановки проблемы. Для этих целей определяется соответствующее информационное пространство, которое обеспечивает технологию постановки проблемы, выявляет области решения и априорную оценку получения результата. Основными составляющими автоматизированной информационной системы являются иерархические, взаимосвязанные база знаний и банк данных. Именно в создании гибкой, открытой, развивающейся базы знаний и ее взаимосвязи с банком данных определена в данных работах основная стратегия построения АИС. Для таких компонентов информационной системы, как формирование запросов пользователей, обработка информации, используются типовые решения. Предлагаемая АИС относится к информационным системам принятия решений. Характерными особенностями такого типа систем являются довольно сложные запросы, с помощью которых производится отбор и анализ данных в различных аспектах: временных, по предметным областям и по другим различным показателям. Для них характерно также наличие средств обработки данных, реализующих функции:

- многомерного анализа данных;
- обработки статистики;
- моделирования и прогноза поведения процессов;
- линейной и нелинейной оптимизации;
- графического представления результатов анализа;
- искусственного интеллекта.

В настоящей работе продолжены исследования по развитию методов и средств построения АИС. Рассмотрена структура и функции АИС, вопросы структуризации информации, технология функционирования

АИС, современные подходы и программные средства построения системы, приведены практические примеры создания информационных систем.

## 1. Структура и функции автоматизированной информационной системы (АИС)

Разработанные структурно-функциональные типы и системные требования к базе знаний и банку данных по технологическим процессам [6-8] позволили создать принципы построения и типовые алгоритмы функционирования автоматизированной информационной системы как открытой, гибкой и развивающейся системы, обеспечивающей решение многофункциональных задач независимо от вида и характера возникающих вопросов.

Основное назначение автоматизированной информационной системы – обеспечение непрерывного сервисного обслуживания процесса принятия решения.

Автоматизированная информационная система реализует функции ввода, контроля и корректировки данных информационной базы, функции хранения, поиска и выборки этих данных по заданным логическим условиям, функции операционной обработки выбранных данных и функции вывода полученных результатов в заданных типовых форматах на устройства отображения информации (дисплей, принтер). Система предусматривает обработку единичных и групповых запросов в режиме монопольного доступа и в режиме мультидоступа к информационной базе. Соответственно в состав АИС входят следующие подсистемы:

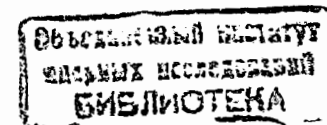
- подсистема ведения классификаторов;
- подсистема ведения входных массивов;
- подсистема обработки и выдачи информации.

Общие принципы построения автоматизированной информационной системы можно сформулировать следующим образом:

- системный подход;
- иерархическое построение;
- модульность;
- технологическая поэтапность;
- способность к адаптации и модификации.

Системный подход, предполагая объединение составных элементов системы по функциональным и информационным признакам, обеспечивает организацию эффективного взаимодействия множества элементов автоматизированной информационной системы.

Иерархическое построение системы (рис.1) ограничивает связь между ее элементами, что существенным образом упрощает и ускоряет



процесс разработки системы и улучшает ее технологические параметры (увеличивает способность к расширению системы, ее модификации и модернизации, упрощает процесс взаимодействия элементов системы и т.д.).

Модульность системы, накладывая на каждый элемент автоматизированной информационной системы требования структурной замкнутости, монолитности и функциональной определенности, позволяет проектировать и разрабатывать любой компонент сложной системы независимо от других компонентов, связанных с ним. Такое построение автоматизированной информационной системы дает целый ряд преимуществ, наиболее важными из которых являются следующие:

- возможность развития системы;
- гибкость системы и ее совместимость с другими системами;
- легкость и простота обслуживания системы.

## 2. Классификационные признаки информации

Прежде чем переходить к более детальному описанию автоматизированной информационной системы, необходимо ввести одно понятие, которое существенно необходимо для ее построения – это условная единица информации (УЕИ) [1]. С использованием данного понятия происходит построение классификаторов, а также описание самого объекта или процесса. Условная единица информации по элементарному объекту (ЭО) содержит две части: типовые составляющие технико-экономических параметров и типовую структуру координат (адресную часть). Структура информации по элементарному объекту может быть представлена в виде постоянной части, отражающей его координаты (адресная часть), и переменной части, отражающей комплекс технико-экономических параметров (рис. 2). Адресная часть должна быть обязательно постоянной на период функционирования объекта, так как координаты элементарных объектов могут определяться однозначно, если множество координат будет иметь конечную размерность. Комплекс технико-экономических параметров, из-за различной степени проработки или изученности процесса или необходимости дополнительных данных по отдельным содержательным аспектам, представляет переменный состав аспектов и соответствующих параметров.

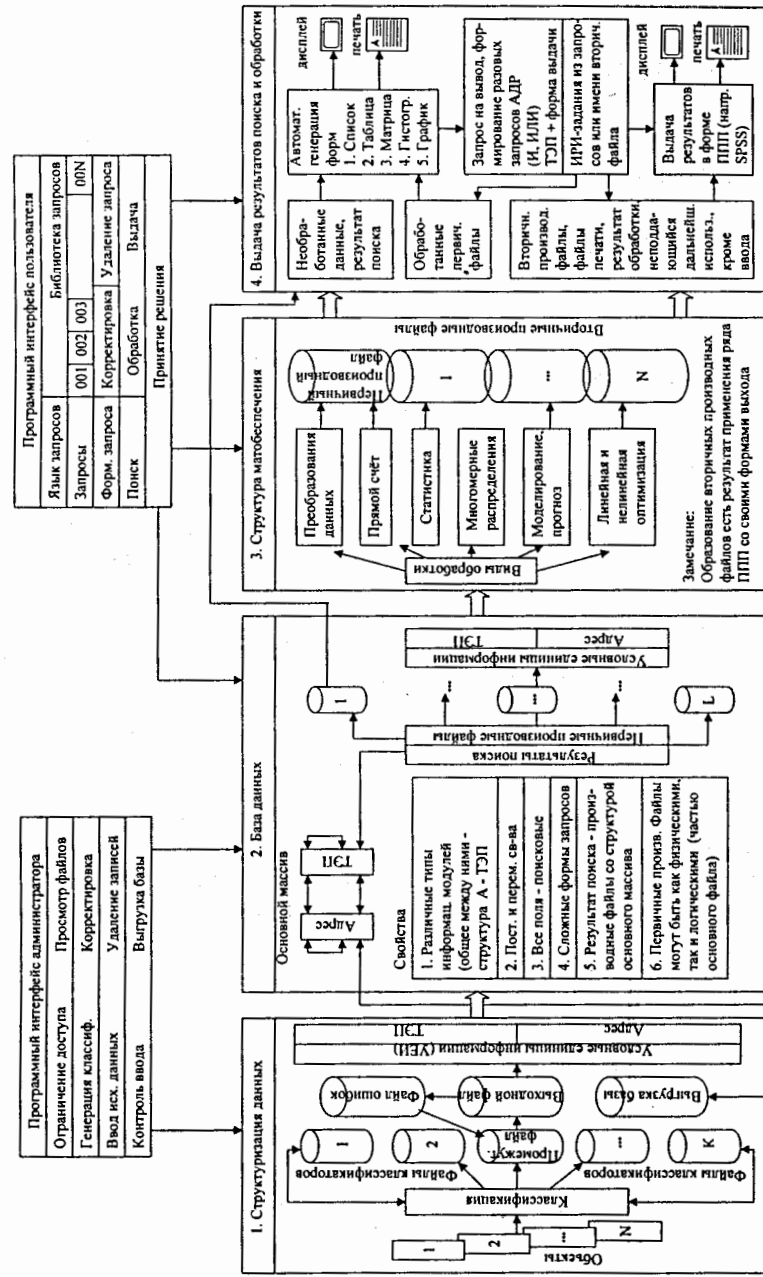


Рис.1. Структура автоматизированной информационной системы (АИС)

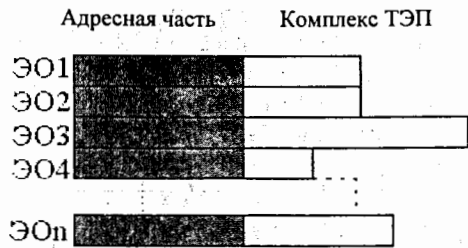


Рис. 2. Структура информации элементарных объектов

Наличие двух частей в структуре условной единицы информации позволяет решать широкий спектр вопросов, связанных с формированием и функционированием базовой и временной информации модели отображения конечного результата. Как уже отмечалось ранее, структура кодов параметров остается неизменной при решении любой задачи, а переменным является их качественное или количественное значение. Качественные и количественные параметры могут быть простого и сложного вида. Параметр простого вида отражает одну из качественно-количественных характеристик элементарных объектов. Например, модуляция – амплитудная, скорость – угловая и т.д. Параметр сложного вида представляет упорядоченное по иерархическому принципу множество понятий, объединенных на каждом уровне единством отражаемых свойств (рис. 3).

Процедура системного анализа «измерение» обеспечивает структуризацию компонентов «вход», «процесс», «выход», информационное наполнение элементарных объектов этих компонентов и является обязательной процедурой для реализации процедур «оценка» и «принятие решения». Недостаточное внимание к процедуре «измерение» и, в первую очередь, структуризации компонентов может привести к значительным ошибкам в реализации процедур «оценка» и «принятие решения».



Рис. 3. Примеры структур параметров сложного вида

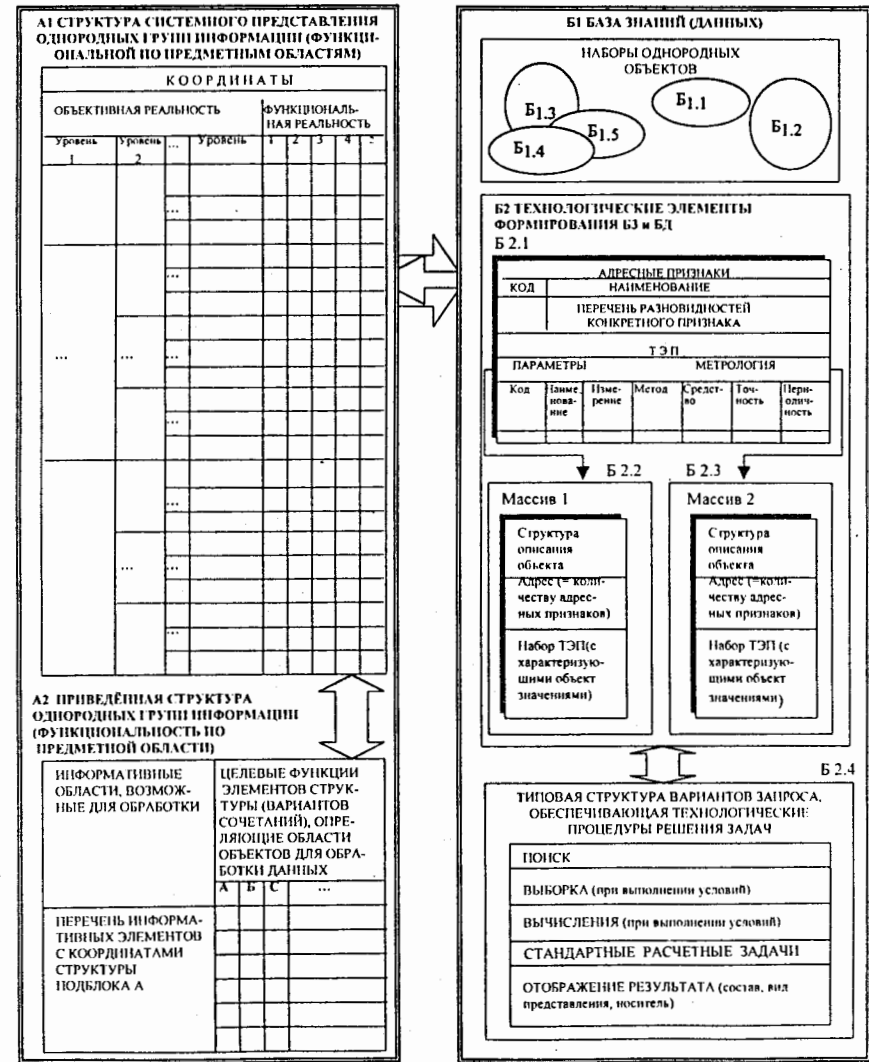


Рис. 4. Блок-схема функционального взаимодействия технологических процедур в автоматизированной информационной системе

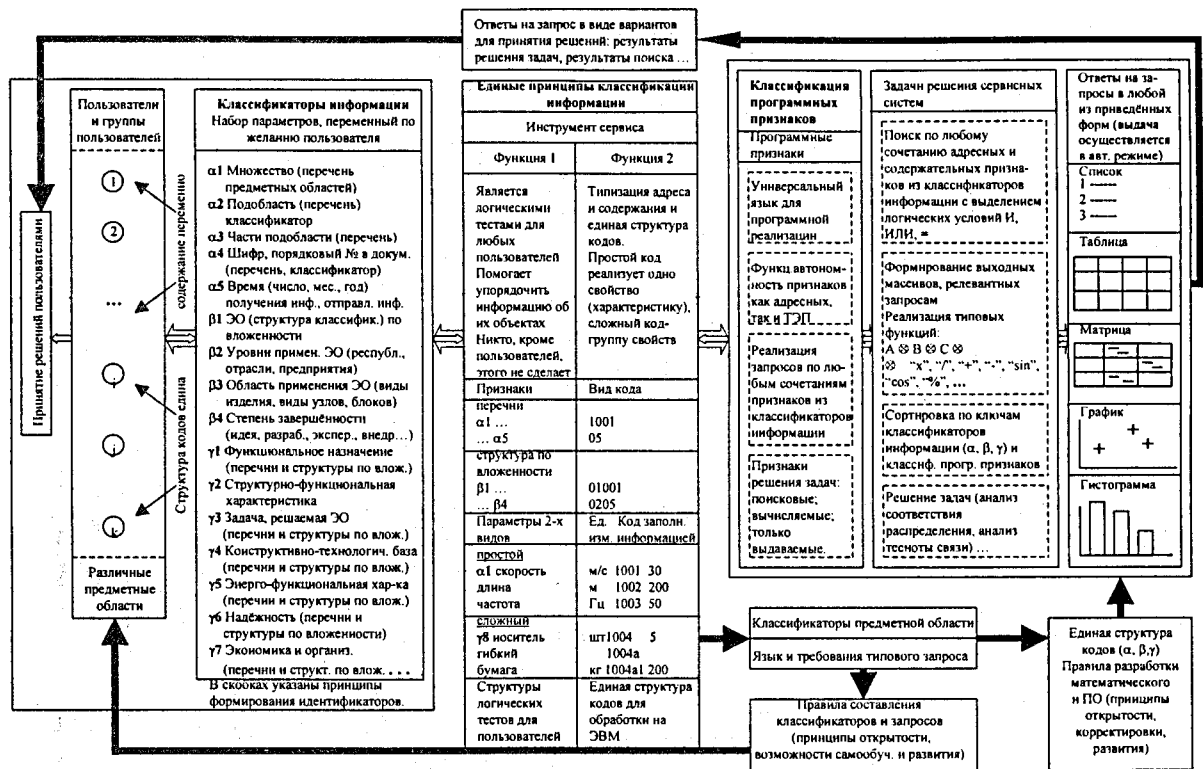


Рис. 5. Сервисная система для разработки и решения широкого класса задач, расширяющая функции и возможности пользователей

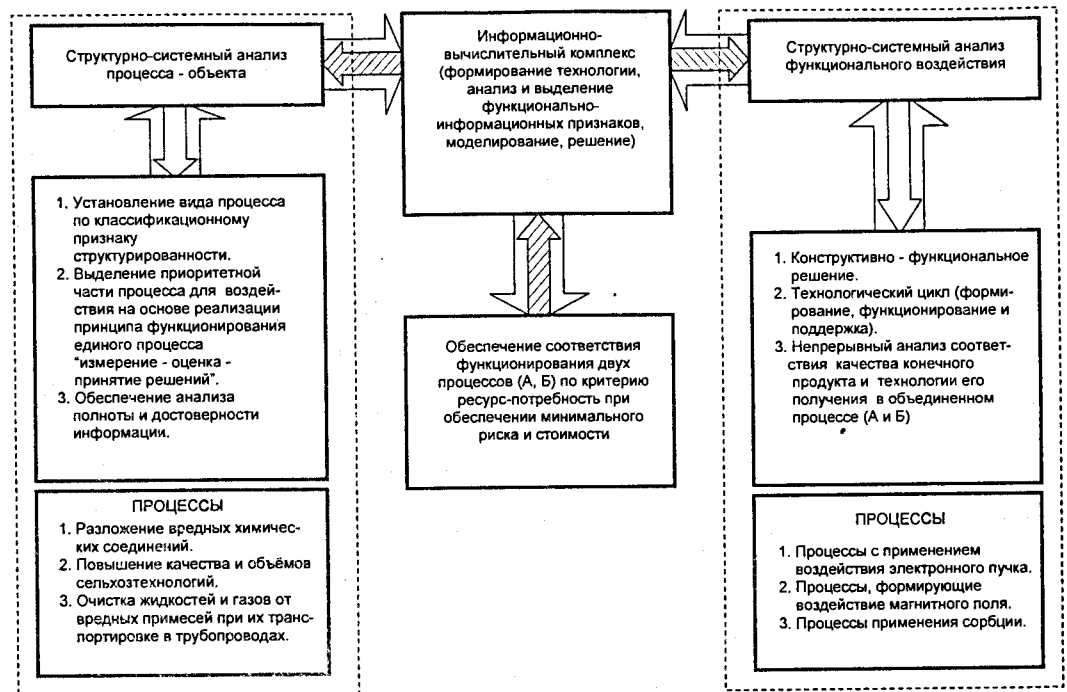


Рис. 6. Основные составляющие технологии системного моделирования для сложных развивающихся технологических процессов

В качестве основного технологического модуля в автоматизированной информационной системе предлагается условная единица информации, которая должна обеспечивать две функции:

1) функцию непрерывной структуризации, подготовки и обработки информации с точки зрения адекватного отражения физической сути информации (рис. 4, блок А) (функция для пользователя) [5];

2) функцию обработки информации с точки зрения технологического процесса получения результата (рис. 4, блок Б) [5].

Автоматизированная информационная система функционирует на информационной базе, построенной по принципам классификации информации. Представление данных в системе и набор процедур обработки данных позволяет использовать автоматизированную информационную систему для решения широкого круга информационных, информационно-логических и расчетных задач (рис. 5).

Классификация информации предполагает однозначное представление информации о каждом объекте (в дальнейшем элементарный объект или ЭО) в виде совокупности информационных признаков, разделенных на две составные части: адресную и технико-экономическую. В адресную часть входят информационные признаки, определяющие координаты элементарного объекта в пространстве и времени, а также характеристики его использования.

Типовая структура координат элементарных объектов, представляющая собой его адресную часть, была получена в результате статистического анализа информации ряда отраслей (табл. 1).

**Таблица 1. Структура адресов элементарного объекта**

Наименование	Обозначение	Содержательные аспекты
Множество ЭО	$\alpha_1$	Аспект, отражающий принадлежность ЭО к определенному множеству (например, для административно-хозяйственной организации - НИИ и КБ; для изделия — завод; для технологического процесса — предприятие, цех и т.д.)
Физическая сущность множества	$\alpha_2$	Аспект, отражающий принадлежность ЭО к одному из объективно существующих научно-технических направлений (например, экономика, радиолокация и т.д.)
Составная часть физической сущности (научно-техническое поднаправление)	$\alpha_3$	Аспект, отражающий принадлежность ЭО к одному из поднаправлений (специализаций) научно-технического направления, указанного в аспекте (например, для экономики — рыночные отношения).

Номер ЭО	$\alpha_4$	Аспект, отражающий ЭО в перечнях, документах.
Год	$\alpha_5$	Аспект, отражающий время начала функционирования, создания, выпуска ЭО.
Вид ЭО	$\beta_1$	Аспект, отражающий принадлежность ЭО к одной из групп и разновидностей ЭО согласно перечню ЭО (НИИ, заготовка, комплекс и т.д.).
Уровень использования	$\beta_2$	Аспект, отражающий возможность использования ЭО в международном, государственном, межотраслевом, отраслевом масштабах и в масштабе предприятия.
Область применения	$\beta_3$	Аспект, отражающий укрупненную характеристику области применения ЭО (например, для изделия — бытовые, промышленные, специальные и т.д.).
Степень завершенности	$\beta_4$	Аспект, отражающий степень готовности ЭО для его использования (например, для изделия — опытный образец, серийные и т.д.)

Девять аспектов координат элементарного объекта в процессе совершенствования информационного отображения объекта могут дополняться и в зависимости от цели отображения объекта использовать либо все аспекты координат, либо некоторые из них, либо определенные их сочетания.

Технико-экономическая часть (табл. 1) представляет собой совокупность технико-экономических параметров, разделенных на восемь основных групп, каждая из которых отражает набор определенных технико-экономических характеристик элементарного объекта:

- функциональное назначение;
- структурно-функциональные характеристики;
- вид задач, решаемых элементарным объектом, при выполнении им функционального назначения;
- характеристики, определяющие конструктивно-технологическую базу;
- энерго-функциональные характеристики;
- эксплуатационные характеристики;
- характеристики надежности функционирования;
- экономические характеристики создания и эксплуатации.

Совокупность технико-экономических параметров в целом отражает как характеристики отдельного экономического объекта, так и характеристики взаимосвязей и взаимодействий различных ЭО. Как видно из таблицы 2, число составляющих (аспектов) — восемь, которое не следует понимать как окончательное. В процессе совершенствования информационного отображения объекта число аспектов может быть увеличено или уменьшено. Состав аспектов при информационном



отображении ЭО может быть использован не полностью, например, если речь идет о структурированном процессе, когда нет необходимости использовать все аспекты, а интерес представляет либо один из них, либо любое их сочетание. В соответствии с разработанным стандартом обеим частям информационных признаков соответствуют единые классификаторы, которые могут составляться как для конкретных экономических объектов, так и для отдельных групп или видов. Степень детализации информации об ЭО для признаков каждого вида может быть различной. В каждом конкретном случае степень детализации информации определяется уровнем использования информации и ограничениями на объем классификаторов. Поскольку структура классификаторов по отдельным составным частям информационных признаков остается неизменной, в процессе эксплуатации системы возможно пополнение и расширение классификаторов.

Таблица 2. Структура технико-экономических параметров элементарных объектов

Наименование	Обозначение	Содержательные аспекты
Функциональное назначение	$\gamma_1$	Аспект, отражающий функциональное назначение ЭО (например, для ЭВМ: ввод-вывод информации, переработка информации и т.п.)
Структурно-функциональная характеристика	$\gamma_2$	Аспект, отражающий структурно-функциональную характеристику ЭО (например, для ЭВМ – конструкция, вид носителя информации, система счисления и т.д.)
Задача, решаемая ЭО	$\gamma_3$	Аспект, отражающий вид задачи, решаемой ЭО, обеспечивающий выполнение им функционального назначения (например, для ЭВМ: буквенно-цифровая печать, графическое представление и т.д.)
Конструктивно-технологическая база	$\gamma_4$	Аспект, отражающий конструктивно-технологическую базу, на которой основано построение ЭО (например, для ЭВМ – микромодули, диоды, литье под давлением, печатный монтаж, стеклотекстолит и т.д.)
Энерго-функциональные характеристики	$\gamma_5$	Аспект, отражающий количественные характеристики функционирования ЭО (например, для ЭВМ – число команд, время выполнения операций, и т.д.)
Эксплуатационные характеристики	$\gamma_6$	Аспект, отражающий эксплуатационные характеристики ЭО (например, для ЭВМ – температура окружающей среды, влажность, давление, габаритные размеры, масса и т.д.)

Надежность	$\gamma_7$	Аспект, отражающий характеристики надежности функционирования ЭО (например, для ЭВМ – наработка на отказ, время восстановления, количество сбоев при считывании и т.д.)
Экономика и организация	$\gamma_8$	Аспект, отражающий экономические характеристики создания и эксплуатации ЭО (для ЭВМ – себестоимость, цена, капитальное вложение, прибыль и т.д.)

#### 4. Структурные характеристики технологии функционирования АИС

В соответствии с требованиями к построению автоматизированной информационной системы она должна реализовывать выполнение следующих процедур: «измерение», «оценка», «принятие решения».

По процедуре «измерение» автоматизированная информационная система обеспечивает формирование классификаторов (адресной и содержательной частей), перечней элементарных объектов и перечня структурных преобразований компонентов «вход», «процесс», «выход».

По процедуре «оценка» автоматизированная информационная система обеспечивает формирование, ведение и корректировку массивов, проведение расчетов по выявлению тесноты взаимосвязей параметров и их группированию [2], поиск информации по любому сочетанию адресных и содержательных аспектов, а также математические действия над параметрами.

Задачами автоматизированной информационной системы по процедуре «принятие решения» являются обеспечение минимального состава классификаторов, возможность получения по запросам потребителей данных по любому сочетанию адресных и содержательных аспектов.

Таким образом, автоматизированная информационная система, с одной стороны, должна обеспечивать выполнение задач, выдвигаемых процедурами «измерение», «оценка», «принятие решения», а с другой стороны, является источником информации для формирования моделей процессов различного уровня и назначения.

Информационные модули (параметр, признак, показатель) и типовые методы решений позволяют выделить четыре вида модулей автоматизированной информационной системы:

- структурно-информационный модуль, включающий информационную модель (все компоненты модели, элементарные объекты этих компонентов или их части), и параметры, т.е. информационное описание объекта;
- информационно-содержательный модуль, включающий информационную модель, параметры и показатели, т.е. данные об объекте или его частях;

- модуль анализа, включающий информационную модель и (или) параметры, и (или) показатели и признаки, т.е. условие задачи анализа, что, в свою очередь, определяет состав пакета прикладных программ по анализу;

- модуль решения типовых задач, включающий типовую структуру входной и выходной информации, пакет прикладных программ по решению типовых задач.

Таким образом, автоматизированная информационная система является гибкой развивающейся системой, технология обработки которой, включает три взаимосвязанных подсистемы (рис. 6) [5]. Взаимосвязь всех элементов трех подсистем осуществляется единым типовым запросом (табл. 3). Как было указано выше, необходимость типового запроса такого вида связана с возможностями принципов классификации информации. Иными словами, типовой запрос является основным инструментом для пользователя систем, а также единой структурой свойств машиноориентированного языка и технологии обработки.

В таблице 4 показан пример применения автоматизированной информационной системы для формирования базы знаний и банка данных системного моделирования технологических процессов.

Таблица 3. Типовой запрос для автоматизированной информационной системы

Смысловое содержание	Элементы запроса		Примечания
	Указывается	Представляется	
Что делать?	Режим работы	Режим работы системы (согласно языку запроса)	Обязательный элемент запроса
Где?	Массивы	Имена массивов, присвоенные им при формировании (согласно языку описания)	Обязательный элемент запроса. Имя массива исх. информации указывается всегда. Имена классификаторов – в зависимости от вида выходной формы
При каком условии?	Поисковые признаки	Каждый признак (согласно языку описания) с соответствующим условием его выборки и правилом поиска (согласно языку запроса)	Возможный элемент запроса
Что вычислять?	Вычисляемые признаки	Каждый признак с соответствующей формулой и способом его вычисления, а для групповых вычислений дополнительно уровни их вычислений	Возможный элемент запроса

Что выдавать?	Выдаваемые признаки	Состав-перечень признаков (согласно языку описания), последовательность указания которых соответствует вложенности признаков (структуре выходной формы) (согласно языку запроса)	Обязательный элемент запроса
Как выдавать?	Выходная форма, ее вид	Выходная форма и вид ее представления (согласно языку запроса)	Обязательный элемент запроса
С каким именем?	Заголовок (имя) ответа на запрос	Имя в зависимости от режима работы может задаваться согласно языку описания (формирование нового массива) или на естественном языке в произвольной форме (режим – отображение)	Возможный элемент запроса
Что дополнительно подключить?	ППП, программа и т.д., программно сопряженные с АИС	Имя ППП, программы и т.д. (согласно языку описания)	Возможный элемент запроса

Таблица 4. Процедуры подготовки исходной информации для создания базы знаний и банков данных

Процедуры	Что определяет	Результат	Примечание
1. Систематизация объекта	Объективная реальность с точки зрения физики предметной области и функциональные аспекты жизнедеятельности	Система координат представления информации	Первичная структуризация информации (с точки зрения физики)
2. Дифференциация систематизированной информации с целью определения области объектов	Полнота представления информации	Выделяются области однородных объектов с точки зрения полноты информационного представления	Возможная структуризация информации
3. Выделение конструктивного (ных) вида(ов) однородных объектов	Целевые функции	Конкретная(ные) область(и) объектов	



4. Детализация и формализация информации выделенных областей	Нормативно-правовое обоснование	Перечень параметров с ед. изм., упорядоченный в соответствии с функц. аспектами жизнедеятельности	Дополнительная структуризация информации
5. Определение типов параметров	Природа возникновения их значений	Дополнение параметров новыми атрибутами в зависимости от значения	Метрологические атрибуты
6. Определение возможных технологических путей обработки конкретных параметров в зависимости от типа его значения	Необходимости значений отдельных атрибутов по каждому признаку	Допустимое использование (обработка) признака в зависимости от полноты представления значений по каждому его атрибуту и, как следствие, достоверность и правомерность использования полученного результата	Один из элементов в цепи шагов принятия решения
7. Определение типов однородных объектов	Цели и решаемые задачи, уровень использования информации (пользователь)	Объект - условная единица информации. Множество однородных объектов - один массив	Элементы структуры БД
8. Выделение признаков, составляющих "адрес" объектов	Принципы классификации информации, согласно которым это могут быть признаки, представляющие пространственно-временные характеристики общего состояния и применения и т.п.	Перечень признаков, определяющих адрес объектов	Может быть один признак. Количество признаков, определяющих адрес, должно быть таковым, чтобы при формировании массива однородных объектов соблюдалось правило: адрес каждого из объектов рамках этого массива – уникален

9. Детальная проработка связей как между признаками, характеризующими объект, так и между массивами	Сформирована система координат и перечень параметров, решаемые задачи для достижения цели	Общие (связующие) признаки, позволяющие максимально реализовать как традиционно сложившийся, реально-объективный набор действий при обработке информации, так и специфику рассматриваемой информации, признаки, позволяющие осуществлять связь между массивами.	
10. Составление локальных классификаторов	Полученный перечень адресных признаков	Уточненная структура и состав каждого классификатора	Схема классификаторов, элементы структуры БД
11. Представление описания объекта в зависимости от ТЭП	Постоянный или переменный состав ТЭП	Состав ТЭП постоянен — табличная форма. Состав ТЭП переменный — по каждому объекту последовательно указывается адрес, затем ТЭП со своими значениями	Классификатор ТЭП можно формировать параллельно с процессом описания объектов. Окончательная структура БД
12. Сбор информации об объектах	Форма сбора	Исходная информация для ввода для автоматизированной обработки	

## 5. Технология создания АИС

Важнейшим решением, принимаемым при создании информационных систем, является выбор и обоснование технологии разработки системы. Правильный выбор позволит адекватно решить поставленную задачу с оптимальными затратами. Технология создания информационных систем представляет собой совокупность методологии построения программных систем и практических инженерных знаний по

организации и управлению разработкой, применяемых в процессе создания программного обеспечения на протяжении всего его жизненного цикла. В настоящей работе рассматривается только методология создания системы, так как принципы организации процесса разработки программных средств обычно определяются группой специалистов, непосредственно реализующих проект системы.

На сегодняшний день существуют два основных подхода к разработке программных систем, различие между которыми обусловлено критериями декомпозиции. Первый подход называют структурным, и в его основу положен принцип функциональной декомпозиции, при которой выделяют функциональные элементы системы и устанавливают строгий порядок происходящих действий. Вторым, объектно-ориентированным подходом опирается на объектную декомпозицию [20]. В этом случае выделяются объекты, содержащие как данные, так и методы их обработки. Рассматриваемые в [1] принципы иерархической структуры построения базы знаний и банка данных с обратными связями очень хорошо соответствует объектному программированию. Создание отражений элементов предметной области в виде объектов – один из эффективных путей упрощения разработки системы. Авторы считают, что именно объектно-ориентированный подход имеет большую перспективу.

Следует отметить, что в будущем АИС должна быть распределенной системой, объединяющей множество баз знаний из различных предметных областей. Хотя на первых порах, скорее всего, будет создана система, включающая единую базу знаний. Новые тенденции – распределенные вычисления – как один из вариантов архитектуры клиент-сервер, позволяют создать интегрированную среду, которая в рамках вычислительной системы не зависит от аппаратных и системных программных средств; опирается на международные и промышленные стандарты; обеспечивает расширяемость системы, т.е. простоту и легкость добавления новых компонентов в существующие системы; позволяет интегрировать старые функционирующие приложения в новые создаваемые вычислительные комплексы; обеспечивает безопасность, надежность и отказоустойчивость; позволяет накапливать, тиражировать и развивать формализованные знания специалистов; существенно снижает суммарные затраты на создание информационных систем. Все это соответствует тем принципам построения автоматизированной информационной системы, которые сформулированы в разделе 2.

В настоящее время появляется множество работ, касающихся описания продуктов, методологий, посвященных созданию распределенных приложений в системе «клиент-сервер» [3, 11], в частности, разработке промежуточного программного уровня (middleware), при помощи которого запросы принимаются от клиентов и направляются соответствующему серверу (технологии CORBA, DCOM [11]).

Распределенная обработка сама по себе не представляет отдельной отрасли техники. Она объединяет ряд таких важных направлений, как распределенные объектные вычисления, техника вычислительных сетей, управление базами данных, протокольная технология. Подобное объединение позволяет создавать коммуникативные, гибкие, надежные и эффективные информационно-вычислительные системы. Однако внедрение таких средств в проектирование и разработку автоматизированных систем физических экспериментов и различных технологических процессов требует преодоления известного консерватизма систем обработки данных в отношении использования новых технологий.

Ниже рассмотрены современные методы и средства создания распределенных информационных систем с целью использования их при решении вопросов создания АИС.

### 5.1. Архитектура взаимодействия компонентов информационной системы

Под архитектурой взаимодействия компонентов информационной системы понимается состав базовых компонентов, их интерфейсы, а также правила и принципы взаимодействия этих компонентов. Проектирование архитектуры взаимодействия компонентов ИС – один из наиболее важных и сложных этапов, и ему не всегда уделяется достаточно внимания при разработке системы.

При проектировании архитектуры взаимодействия распределенных компонентов информационной системы различают следующие типы взаимодействия [12]:

- вертикальный – каждый компонент имеет уникальный в рамках данной информационной системы интерфейс;
- горизонтальный – все компоненты имеют один и тот же универсальный интерфейс, обеспечивающий межкомпонентное взаимодействие;
- смешанный – все компоненты имеют универсальный базовый интерфейс, при этом каждый компонент специфицирует дополнительные операции для работы со своим доменом предметной области.

Достоинства и недостатки этих типов взаимодействия легко проследить на примере интеграции новой задачи в информационную систему, состоящую из нескольких компонентов.

В случае использования вертикального типа взаимодействия распределенных компонентов объем кода, необходимого для интеграции, будет определяться количеством компонентов системы.

Количество интерфейсов в этом случае будет порядка  $N*N$ , где  $N$  – число компонентов. В случае горизонтально построенной системы

количество интерфейсов (интеграционного кода) будет минимальным. Добавление нового компонента потребует реализовать дополнительно всего два интерфейса.

При реальной разработке чрезвычайно трудно создать одинаковые интерфейсы для всех подсистем, поэтому наиболее предпочтительным для создания распределенной системы является смешанный тип взаимодействия компонентов. В этом случае построенная архитектура сохраняет свойство универсальности интерфейсов, но позволяет добавлять специфичные для данного приложения операции. Количество кода, необходимого для интеграции новой задачи при такой архитектуре, изменяется в зависимости от конкретного проекта от 2 (горизонтальная архитектура) до  $2N+1$  (вертикальная архитектура). Общие интерфейсы компонентов – ключ к проектированию архитектуры, обеспечивающей успешное развитие и поддержку системы на протяжении длительного периода.

При проектировании архитектуры приложений первым шагом должен быть выбор используемых стандартов. С увеличением сложности информационных систем важность соответствия программного обеспечения стандартам возрастает. Стандарты используются для достижения следующих целей:

- портируемость приложений – перенос приложений на различные аппаратные платформы, операционные системы, сетевые протоколы;
- интероперабельность (interoperability) – возможность совместного использования информации и ресурсов компонентами распределенной системы [13];
- снижение стоимости системы – интеграция программных систем, поддерживающих общепринятые стандарты, уменьшает стоимость приложений для конечного использования;
- снижение риска выбора программного продукта – использование стандартов освобождает разработчика от привязанности к конкретному программному продукту;
- увеличение времени жизненного цикла системы – соответствие стандартам уменьшает риск быстрого устаревания системы.

## 5.2. Функциональная нагрузка компонентов системы

Автоматизированная информационная система состоит из совокупности взаимодействующих друг с другом программных компонентов. Каждый из таких компонентов представляет собой программный модуль, исполненный в рамках отдельного процесса. В компоненте, согласно [3], могут быть реализованы функции из трех групп:

- 1) группа функций пользовательского интерфейса;

- 2) группа прикладных функций, характерных для данной предметной области;
- 3) группа функций хранения и управления данными.

Декомпозиция функций из этих групп по компонентам определяет логическую архитектуру информационной системы [11]. С каждой логической архитектурой может быть связана диаграмма, описывающая отношения взаимодействия компонентов в ИС.

Такая диаграмма представляет собой ориентированный граф. Дуги графа определяют отношения взаимодействия между программными компонентами ИС, являющимися его вершинами.

Выделяют следующие, наиболее распространенные классы логических архитектур:

- двухуровневые архитектуры;
- трехуровневые;
- распределенные архитектуры (существуют отношения взаимодействия между компонентами, каждый из которых реализует функции всех трех групп).

### 5.2.1. Двухуровневые архитектуры

Двухуровневая архитектура (или двухуровневый “клиент-сервер”) означает наличие максимум двух программных компонентов, в совокупности реализующих функции из первой, второй и третьей группы.

Один из двух компонентов выступает в роли сервера, т.е. реализует набор сервисов, доступных другому компоненту, который выступает в роли клиента, т.е. в процессе работы пользуется сервисами, предоставляемыми сервером. Компоненты могут располагаться как на одном компьютере, так и на различных компьютерах, объединенных в сеть.

Различия в реализациях двухуровневой архитектуры определяются в основном тем, функции каких групп выполняет клиент, а каких сервер. Существует несколько вариантов декомпозиции функций представления, прикладных функций и функций хранения и управления данными в рамках двухуровневой архитектуры ИС.

Согласно указанным вариантам декомпозиции, можно говорить о следующих двухуровневых архитектурах:

- интеллектуального клиента (доступ к удаленным данным, архитектура на базе файлового сервера);
- интеллектуального сервера (доступ в режиме терминала, архитектура интеллектуального сервера баз данных);
- распределенной функциональной логики.

Двухуровневые архитектуры обладают рядом достоинств и недостатков, частично описанных в [3, 14]. Использование двухуровневых

архитектур при построении крупных информационных систем, исходя из присущих им недостатков, приводит во многих случаях к краху проектов, связанных с их разработкой.

### 5.2.2. Трехуровневые архитектуры

Трехуровневая архитектура (или трехуровневый «клиент-сервер») предполагает наличие максимум трех программных компонентов, которые участвуют в процессе, предполагающем выполнение функций из первой, второй и третьей групп.

Трехуровневые архитектуры предусматривают не столь жесткие связи между клиентом и сервером и более гибкие формы распределенной обработки [14]. Наиболее распространенной считается архитектура, согласно которой выделяются три компонента ИС (представления, прикладной, доступа к информационным ресурсам), являющиеся автономными и общающиеся через средства межпроцессорного взаимодействия при помощи стандартных интерфейсов.

Отдельные компоненты могут располагаться как на одном компьютере, так и на разных компьютерах, обеспечивая тем самым распределенную обработку информации. Компонент представления часто располагается на персональном компьютере, прикладной компонент (называемый также сервером приложения) выполняется сервером среднего уровня под управлением операционной системы Unix или Windows NT, а компонент доступа к данным и сами данные располагаются либо на мощных Unix-серверах, либо на больших или мини-ЭВМ. Однако на практике все три компонента могут с успехом выполняться и на одном компьютере.

Основным элементом трехуровневой архитектуры является сервер приложения. Как правило, в нем реализуется несколько прикладных функций, каждая из которых оформлена как сервис (service) и предоставляет некоторые услуги всем компонентам представления, которые желают и могут ими воспользоваться. Серверов приложений может быть несколько, и каждый из них может предоставлять определенный набор сервисов. Детали реализации прикладных функций в серверах приложений полностью скрыты от клиентов. Кроме того, разработчики могут создавать, изменять или переносить любые компоненты ИС, практически не затрагивая других.

### 5.2.3. Распределенные архитектуры

Согласно распределенной архитектуре клиент [4, 14], взаимодействующий с сервером приложений, трактуется более широко, чем компонент представления. Он может поддерживать интерфейс с конечным пользователем, а может выполнять прикладные функции и являться сервером приложения. В общем случае, в рамках данной

архитектуры, клиент (сервер) может как предоставлять, так и запрашивать некоторые сервисы. Это позволяет на этапе проектирования информационной системы осуществить такую декомпозицию функций из указанных выше трех групп по компонентам ИС, которая была бы оптимальной в контексте решаемой задачи.

Для обеспечения взаимодействия компонентов информационной системы, поддерживающей распределенную архитектуру, необходимо создать промежуточный программный уровень (middleware), при помощи которого запросы принимаются от клиентов и направляются соответствующему серверу. Сегодня уже имеется или анонсировано достаточное количество инструментальных средств, позволяющих разработчикам строить распределенные ИС, не вдаваясь в детали реализации взаимодействия клиента и сервера. Многие из этих программных продуктов реализуют стандарт CORBA (Common Object Request Broker Architecture), DCOM и будут рассмотрены в разделе 5.4.

Распределенные архитектуры имеют более универсальный характер, чем двухуровневые и трехуровневые архитектуры. Четкое разграничение логических компонентов, присущее распределенным архитектурам, и рациональный выбор программных средств позволяют достичь такого уровня гибкости, открытости и производительности ИС, который пока недостижим при использовании двухуровневых и трехуровневых архитектур.

### 5.3. База данных в распределенных информационных системах

База данных является основным элементом информационной системы. В распределенных системах качество использования сервера базы данных зависит от его места в модели технологии «клиент-сервер».

Различия в реализациях технологии «клиент-сервер» определяются четырьмя факторами [3]. Во-первых, в какие виды программного обеспечения интегрирован каждый из этих компонентов. Во-вторых, какие механизмы программного обеспечения используются для реализации функций всех трех групп. В-третьих, как логические компоненты распределяются между компьютерами в сети. В-четвертых, какие механизмы используются для связи компонентов между собой.

Выделяются четыре подхода, реализованные в моделях:

- модель файлового сервера (File Server - FS);
- модель доступа к удаленным данным (Remote Data Access - RDA);
- модель сервера базы данных (DataBase Server - DBS);
- модель сервера приложений (Application Server - AS).

Модель файлового сервера FS является базовой для небольших локальных сетей компьютеров и сейчас почти не используется для

создания сложных информационных систем. Последующие три модели широко используются при построении современных информационных систем.

В модели доступа к удаленным данным, RDA-модели, коды компонента представления и прикладного компонента совмещены и выполняются на компьютере-клиенте. Последний поддерживает как функции ввода и отображения данных, так и чисто прикладные функции. Компонентом доступа к информационным ресурсам является, как правило, SQL-сервер. Доступ к информационным ресурсам обеспечивается либо операторами специального языка (языка SQL, например, если речь идет о базах данных), либо вызовами функций специальной библиотеки (если имеется соответствующий интерфейс прикладного программирования - API). В этой модели уже используется унифицированный интерфейс «клиент-сервер» в виде языка SQL. Недостатками RDA-модели считается, во-первых, то, что взаимодействие клиента и сервера посредством SQL-запросов существенно загружает сеть; во-вторых, удовлетворительное администрирование приложений в RDA-модели практически невозможно из-за совмещения в одной программе различных по своей природе функций (функции представления и прикладные).

Все большую популярность приобретает перспективная модель сервера базы данных (DBS-модель). Последняя реализована в некоторых реляционных СУБД (Informix, Ingres, Sybase, Oracle, SQL Server). Ее основу составляет механизм хранимых процедур - средство программирования SQL-сервера. Процедуры хранятся в словаре базы данных, разделяются между несколькими клиентами и выполняются на том же компьютере, где функционирует SQL-сервер. Язык, на котором разрабатываются хранимые процедуры, представляет собой процедурное расширение языка запросов SQL и уникален для каждой конкретной СУБД. В DBS-модели компонент представления выполняется на компьютере-клиенте, в то время как прикладной компонент оформлен как набор хранимых процедур и функционирует на компьютере-сервере БД. Там же выполняется компонент доступа к данным, то есть ядро СУБД. Достоинства DBS-модели очевидны: это и возможность централизованного администрирования прикладных функций, и снижение трафика (вместо SQL-запросов по сети направляются вызовы хранимых процедур), и возможность разделения процедуры между несколькими приложениями, и экономия ресурсов компьютера за счет использования единожды созданного плана выполнения процедуры. К недостаткам модели можно отнести ограниченность средств, используемых для написания хранимых процедур, которые представляют собой разнообразные процедурные расширения SQL, не выдерживающие сравнения по изобразительным средствам и функциональным возможностям с языками третьего поколения, такими, как С или Pascal. Сфера их использования ограничена конкретной СУБД, в большинстве

СУБД отсутствуют возможности отладки и тестирования разработанных хранимых процедур.

На практике часто используются смешанные модели, когда поддержка целостности базы данных и некоторые простейшие прикладные функции поддерживаются хранимыми процедурами (DBS-модель), а более сложные функции реализуются непосредственно в прикладной программе, которая выполняется на компьютере-клиенте (RDA-модель). Так или иначе, современные многопользовательские СУБД опираются на RDA- и DBS-модели, и при создании ИС, предполагающем использование только СУБД, выбирают одну из этих двух моделей либо их разумное сочетание.

В модели сервера приложений (AS-модели) процесс, выполняющийся на компьютере-клиенте, отвечает, как обычно, за интерфейс с пользователем (то есть осуществляет функции первой группы). Прикладной компонент реализован как группа процессов, выполняющих прикладные функции, и называется сервером приложения. Все операции над информационными ресурсами выполняются соответствующим компонентом, по отношению к которому сервер приложения играет роль клиента. Из прикладных компонентов доступны ресурсы различных типов - базы данных, очереди, почтовые службы и др. В AS-модели реализована трехзвенная схема разделения функций, где прикладной компонент выделен как важнейший изолированный элемент приложения, для его определения используются универсальные механизмы многозадачной операционной системы, и стандартизованы интерфейсы с двумя другими компонентами. AS-модель является фундаментом для мониторов обработки транзакций, или, проще, мониторов транзакций, которые выделяются как особый вид программного обеспечения. Пример AS-модели приведен на рис. 7.



Рис. 7. Модель сервера приложений



## 5.4. Сетевые решения в информационной системе

### 5.4.1. Сетевая архитектура

С развитием технологий распределенных вычислений изменялись и требования к вычислительным сетям. Сегодня уже основная задача разработки сетей распределенных систем заключается не в обеспечении связи, а в перемещении больших объемов данных, необходимых для распределенных вычислений [15]. Девиз фирмы «Sun»: «Сеть – это компьютер» отражает современные требования к вычислительным сетям. Представление о сети как о «трубопроводе», по которому передается информация от компьютера к компьютеру, постепенно устаревает. В современных информационных технологиях сеть – это, прежде всего, основа для работы программного обеспечения. Разработчики программного обеспечения вычислительных систем уже не могут не учитывать связи между компьютерами, которые обеспечивает сеть. Современные вычислительные сети не только обеспечивают связь между компьютерами, но и являются основой для распределенных вычислений.

Согласно [1, 16], современная компьютерная сеть должна удовлетворять трем основным требованиям, а именно:

- хорошей масштабируемости,
- высокой производительности,
- управляемости,
- коммуникативности.

Хорошая масштабируемость необходима для того, чтобы можно было изменять, расширять состав программных компонентов, используемых в сети, менять число пользователей, работающих с вычислительной системой. Высокая производительность сети требуется для ускорения работы программ. Масштабируемость теряет свои достоинства при изначально неудовлетворительной производительности. Всегда важно осознавать, что более скоростное аппаратное обеспечение дает вам следующий шаг в развитии. И, наконец, сеть должна быть управляемой, чтобы ее можно было легко перенастроить на новые условия с наименьшими затратами. Кроме того, сетевые соединения не так устойчивы, как соединения внутри машины. Компоненты системы должны знать, что клиент более не активен, даже в случае сетевой или аппаратной аварии. Вопросы коммуникативности обсуждаются в [1].

Эти требования отражают новый этап в развитии сетевых технологий – этап создания высокопроизводительных сетей. При использовании сети необходимо, чтобы программы (и сетевая инфраструктура) могли обеспечивать большое количество операций «клиент-сервер». Если же архитектура сети не предназначена для распределенных вычислений, то система будет работать неэффективно.

Поэтому многие разработчики систем, используя передовые информационные технологии, стремятся добиться максимальной производительности своей сети.

Своеобразие новых программ и технологий усложняет разработку вычислительных систем. Централизованные ресурсы, новые классы программ, новые принципы их применения, изменение информационных потоков, увеличение числа одновременно работающих пользователей и более мощные вычислительные платформы – все эти факторы нужно учитывать при разработке компьютерных сетей.

Создано большое количество технологических и архитектурных решений, и выбрать из них наиболее подходящее – достаточно сложная задача. Основной классификационный признак подразделения сетей – это их масштабность, размеры в пространстве. Локальные вычислительные сети (ЛВС) – это сети с расстоянием между отдельными компьютерами (узлами сети), не превышающим 2,5 – 3,0 км [16]. Глобальные вычислительные сети объединяют компьютеры в больших географических регионах, располагают значительными вычислительными мощностями и позволяют получить доступ к информационным ресурсам всей планеты (например, сеть Internet). Основными транспортными методами ЛВС являются такие технологии, как Ethernet, Token Ring, FDDI [16]. Для больших сетевых расстояний используются сетевые технологии – ISDN, X25, Frame Relay. Из универсальных технологий следует отметить очень перспективную ATM-технологию [19]. Различные сетевые технологии обеспечивают разное время реакции и общей пропускной способности сети [13, 16] (рис. 8).

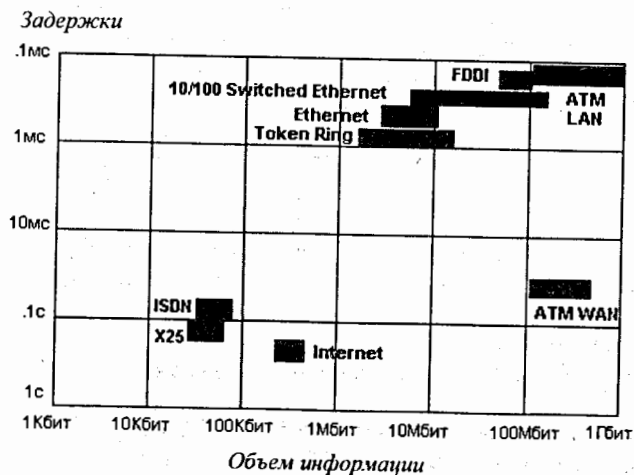


Рис. 8. Производительность основных сетевых технологий



Технологии коммутации кадров и ячеек позволили увеличить пропускную способность сети, дали возможность передавать большие объемы данных за короткое время. Такие технологии, как ATM и Ethernet, обеспечивают пропускную способность сети в диапазоне от 10 до 100 Мбит/с и выше. В ЛВС и глобальной сети диапазон пропускной способности составляет шесть порядков (от 1 тыс. бит/с до 1 млрд бит/с), а сетевых задержек – три порядка (от 1 с до 1 мс).

Требования к сети, обусловленные требованиями разрабатываемой системы, используемым программным обеспечением, очень широки и разнообразны. Общих рекомендаций выбора сетевой архитектуры нет. Вопросы сетевой архитектуры, быстродействия, оптимизации сетевых обменов, безопасности и устойчивости сети к авариям решаются специалистами исходя из конкретных требований к вычислительной системе.

#### 5.4.2. Базовые механизмы сетевых взаимодействий

Гетерогенные вычислительные среды сегодня стали реальностью для многих систем обработки данных. В связи с этим повышаются требования к интеграции разнородных приложений, функционирующих в распределенных средах с широким диапазоном платформ и сетей. Перед разработчиками приложений, во-первых, стоит проблема создания такого программного обеспечения, которое могло бы работать на максимально возможном числе платформ, используемых в организации. Во-вторых, разработка приложений для автоматизированных систем должна осуществляться на основе либо собственных стандартов (замкнутое решение), либо общепринятых международных стандартов. Используя международные стандарты, можно достаточно просто интегрировать в ИС программные продукты фирм, следующих этим стандартам. В-третьих, разработчикам необходимо предусмотреть возможность модификации приложений ИС так, чтобы процесс модернизации был минимален по времени и затратам. А этого можно достичь, следуя определенным принципам выделения и объединения компонентов информационных систем.

Существующие механизмы и технологии интеграции компонентов информационных систем можно разделить на следующие категории [12, 15]:

- 1) частные решения;
- 2) разнообразные (смешанные) механизмы (Miscellaneous Mechanisms);
- 3) удаленные вызовы процедур на базе DCE RPC;
- 4) распределенные объекты (CORBA, DCOM).
- 5) «frameworks»;

б) стандартные архитектуры.

Коротко остановимся на решениях, соответствующих каждой из категорий. Следует отметить, что одним из основных требований, которым должны удовлетворять распределенные информационные системы, является использование программного обеспечения и технологий, согласующихся с общепризнанными стандартами, определяющими принципы взаимодействия компонент АИС.

#### Частные решения

Технологические решения, относящиеся к первой категории, базируются на основе собственных (уникальных для данной организации) протоколов и интерфейсов взаимодействия. Такие решения, в большинстве случаев, порождают непреодолимые трудности при попытке организации общения компонентов данной АИС с компонентами информационных систем, построенных на основе других решений межкомпонентного взаимодействия.

#### Смешанные механизмы

Ко второй категории относятся технологические решения, предполагающие построение информационных систем из расчета на конкретную задачу и изначально не рассчитанные на использование технологий интеграции систем. Поэтому в данном случае в качестве средств интеграции используются такие механизмы, как сокет (sockets) протокола TCP/IP и ONC RPC (Object Network Computing Remote Procedure Call). Программное обеспечение данной категории, как правило, разрабатывается для использования только внутри конкретной организации, что приводит к резкому увеличению затрат при интеграции с другими системами.

Механизм сокет, или механизм программных гнезд, впервые был реализован в 1982 году в UNIX в качестве развитого средства межпроцессных взаимодействий. Это средство позволяет любому процессу обмениваться сообщениями с любым другим процессом, независимо от того, выполняются они на одном компьютере или на разных, соединенных сетью. Функционально механизм сокет близок к возможностям TLI (пятого уровня в соответствии с моделью ISO/OSI).

Сокет входит в число обязательных компонентов таких стандартных сред, как ОС UNIX, MS Windows, однако реализуется в разных системах по-разному. Например, в UNIX-BSD ориентированных системах механизм сокет исторически реализуется в ядре ОС, и пользователям предоставляются пять специальных системных вызовов: socket, bind, listen, connect и accept [14]. В MS Windows сокет поддерживается в виде набора библиотечных функций (библиотеки Winsocket).

Основными идеями механизма вызова удаленных процедур (RPC - Remote Procedure Calls) являются следующие:

а) во многих случаях взаимодействие процессов носит ярко выраженный асимметричный характер. Один из процессов («клиент») запрашивает у другого процесса («сервера») некоторую услугу (сервис) и не продолжает свое выполнение до тех пор, пока эта услуга не будет выполнена (и пока процесс-клиент не получит соответствующие результаты). Видно, что семантически такой режим взаимодействия эквивалентен вызову процедуры, и естественно желание оформить его должным образом синтаксически;

б) свойства переносимости позволяют, в частности, предельно просто создавать «операционно-однородные» сети, включающие разнородные компьютеры. Однако остается проблема представления данных в компьютерах различной архитектуры (часто отличается представление числа с плавающей точкой, используется другой порядок размещения байтов в машинном слове и т.д.). Плохо, когда решение проблемы разных представлений данных возлагается на пользователей. Поэтому второй идеей RPC (многие считают, что это основная идея) является автоматическое обеспечение преобразования форматов данных при взаимодействии процессов, выполняющихся на разнородных компьютерах.

Впервые пакет RPC был реализован компанией «Sun Microsystems» в 1984 году в рамках ее продукта NFS (Network File System – сетевая файловая система) [12]. Пакет был тщательно специфицирован с тем, чтобы пользовательский интерфейс и его функции не были зависимыми от применяемого транспортного механизма. Специальные системные вызовы для реализации RPC не поддерживаются.

### **Удаленные вызовы процедур на базе OSF DCE**

В технологических решениях третьей категории предполагается использование технологий, базирующихся на средствах единообразной интеграции компонентов и имеющих преимущества в сравнении с решениями предыдущей категории. К данной категории относятся технологии на базе механизма вызова удаленных процедур (RPC), такие, как OSF DCE. Как известно, OSF DCE определяет сервисы безопасности, именования и другие важные механизмы, необходимые для интеграции систем в распределенной среде. С другой стороны, создание объектно-ориентированной системы на базе DCE не может быть оптимальным решением в силу не полной объектной ориентированности последней. Характерным недостатком OSF DCE считается сложность создания объектов как независимых компонентов распределенной системы и ориентированность на процедурный стиль программирования. Поэтому имеется тенденция рассматривать DCE как технологию, которая устарела

в сравнении с новыми объектно-ориентированными технологиями построения распределенных систем, такими, как технология CORBA и DCOM (по мере принятия стандарта). Использование данной технологии может привести к моральному износу системы.

### **Распределенные объекты (CORBA, DCOM)**

В четвертой категории при построении информационных систем используются, в основном, две технологии распределенных объектных программных систем – CORBA и DCOM [11, 12]. Тенденции в развитии программного обеспечения ЭВМ, представляющие собой парадигмы: объектно-ориентированный способ разработки [20] и распределенные вычисления, послужили основой созданного в апреле 1989 года консорциума «Object Management Group», членами которого сейчас являются 555 ведущих компьютерных компаний, таких, как «Sun», DEC, IBM, HP, «Motorola» и т.д. Задачей консорциума ставится разработка спецификаций и стандартов, позволяющих строить распределенные объектные системы в гетерогенных средах. Появление такого стандарта должно облегчить процесс внедрения распределенных объектных технологий в разработку приложений вычислительных систем. Необходимая спецификация была разработана OMG и получила название «Object Management Architecture» (OMA). OMA состоит из четырех основных компонентов, представляющих собой спецификации различных уровней поддержки приложений:

- архитектура брокера запросов объектов (CORBA – Common Object Request Broker Architecture) устанавливает базовые механизмы взаимодействия объектов в гетерогенной сети;
- сервисы объектов (Object Services) являются основными системными службами, используемыми разработчиками для создания приложений;
- универсальные средства (Common Facilities) ориентированы на поддержку пользовательских приложений, таких, как электронная почта, средства печати и т.д.;
- объекты приложений (Application Objects) предназначены для решения конкретных прикладных задач.

Спецификация CORBA лежит в основе любого компонента, разработанного OMG. CORBA определяет механизм, обеспечивающий взаимодействие приложений в распределенной среде.

Главные компоненты стандарта CORBA:

- объектный брокер запросов (Object Request Broker);
- язык определения интерфейсов (Interface Definition Language);
- объектный адаптер (Object adapter);
- репозиторий интерфейсов (Interface Repository).

Концептуально CORBA относится к уровням приложений и представлений семиуровневой модели OSI [12]. Она обеспечивает возможность построения распределенных систем и приложений на самом высоком уровне абстракции в рамках стандарта OSI. С ее помощью возможно изолировать клиентские программы от низкоуровневых, гетерогенных характеристик информационных систем. Характерные особенности разработки по технологии CORBA заключаются в следующем:

- язык описания интерфейсов OMG IDL позволяет определить интерфейс, независимый от языка программирования, используемого для реализации;
- высокий уровень абстракции CORBA в семиуровневой модели OSI позволяет программисту не работать с низкоуровневыми протоколами;
- программисту не требуется информация о реальном месте расположения сервера и способе его активации.

Конкурирующей с CORBA распределенной объектной технологией является DCOM (Component Object Model), разработанная корпорацией «Microsoft». Сравнение этих двух архитектур приведено в [11].

#### **Технологии «framework»**

Технологические решения этой категории предполагают разработку «frameworks» как основы программной архитектуры для использования в нескольких проектах. Как правило, «framework» воплощает тщательно продуманные принципы построения программной архитектуры и довольно часто создается с помощью средств, определенных в стандарте CORBA. Разработка «frameworks» – направление многообещающее и бурно развивающееся в последнее время.

#### **Стандартные архитектуры**

К шестой категории относятся высококачественные технологии, программные архитектуры и сервисы, которые рассчитаны на повторное использование во множестве различных проектов. Как правило, такие решения опираются на стандарт CORBA, что в конечном итоге позволяет организациям практически свести на нет риск устаревания систем. Важно отметить, что не каждая организация способна создать технологию мирового класса. Организации, разрабатывающие решения шестой категории, – всемирно известные фирмы, являющиеся создателями стандартов взаимодействия систем в своей отрасли.

#### **5.4.3. Методы описания сетевых услуг и протоколов**

Основой информационного взаимодействия распределенных вычислительных систем, во многом определяющей корректность, надежность и эффективность их функционирования, являются протоколы, разработке которых уделяется большое внимание [10]. Поэтому прежде всего встает вопрос о выборе формы описания протоколов и сервисов.

Первые описания протоколов носили словесный характер. Каждое из таких описаний вводит свои терминологические понятия, которые и используются в дальнейшем. Однако уже в первые годы создания сетей ЭВМ была признана неудовлетворительность словесной формы, что породило множество различных, никак не связанных друг с другом формальных и полужформальных методов [10]. И хотя огромное число быстро возникающих способов, создаваемых для описания протоколов, задержало появление единых средств, они, безусловно, сыграли положительную роль в исследовании и получении значительного опыта в данной области.

В настоящее время наиболее широко используются три основных подхода к формальному описанию указанных спецификаций: расширенная автоматная модель, расширенные сети Петри, расширенные модели состояний. Подробный анализ методов дан в [9, 17, 18].

### **6. Практические примеры построения автоматизированных информационных систем**

#### **6.1. Автоматизированная информационная система учета и контроля ядерных материалов (ЯМ) в ОИЯИ**

Автоматизированная информационная система учета и контроля ЯМ, получившая наименование MTIS (Materials Tracking Information System), предназначена для автоматизированного ведения учета ЯМ, используемых в ОИЯИ, слежения за их перемещениями, изменениями их инвентарных количеств, подготовки требуемой учетной документации, а также для информационной поддержки мероприятий, проводимых в Институте по безопасности хранения и использования ЯМ [21].

Система MTIS ведет учет материалов в виде штучных учетных единиц, которые определены для системы, как условная единица информации. Для описания учетной единицы используется более 40 параметров. Эти параметры разбиты на две группы – параметры описания учетной единицы и параметры описания материалов учетной единицы. К параметрам учетной единицы относятся:

- физическая форма;
- химическая форма;

тип упаковки;  
местоположение  
и др.

Параметрами описания материалов учетной единицы являются следующие характеристики:

тип материала (наименование материала);  
лигатурный вес;  
чистый вес;  
единица измерения веса;  
процентное содержание изотопа;  
активность;  
базис измерения;  
и др.

Для каждого параметра составлены классификаторы, определяющих перечень возможных значений. Например, для химической формы классификатор включает до 20 позиций.

Пользователям MTIS предоставляется удобный инструментарий для построения запросов к базе данных. В системе разработан как типовой набор запросов, используемых пользователями в повседневной работе, так и дополнительно включен редактор запросов, позволяющий строить новые запросы. Результаты запросов выдаются в виде таблиц, отчетов, списков.

В MTIS реализована модель архитектуры "клиент-сервер". Используется модель сервера базы данных для большинства задач системы и модель доступа к удаленным данным для задач редактирования запросов (см. 3.2.). Для написания хранимых процедур модели сервера базы данных использовались средства процедурного расширения языка запросов SQL СУБД Microsoft SQL Server – transact-SQL.

В состав локальной сети системы входят два сервера (основной и резервный) и ряд рабочих станций, в зависимости от количества пользователей. В MTIS используется операционная система Windows NT (Windows NT Server 4.0 и Windows NT Workstation 4.0) и СУБД на базе Microsoft SQL Server 6.5. Эти программные продукты прошли аттестацию на государственном уровне.

## 6.2. Федеральная система учета и контроля ЯМ

Федеральная информационная система учета и контроля ядерных материалов Российской Федерации – это централизованная система сбора, хранения и распределения данных по учету и контролю ядерных материалов, построенная с использованием современных компьютерных технологий [22-24]. Компонентами, составляющими федеральную информационную систему, являются: Федеральный информационный

центр (ФИЦ), ядерные установки, министерства и ведомства России. Информацию в ФИЦ предоставляют ядерные установки. Потребителями информации являются министерства и ведомства России, ядерные установки, поставляющие информацию, а также специалисты-аналитики информационного центра. Структура Федеральной информационной системы приведена на рис.9.



Рис.9. Компоненты Федеральной информационной системы

Разработка проекта и создание Федеральной информационной системы (ФИС) учета и контроля ядерных материалов России предоставили широкие возможности для приобретения практического опыта, необходимого для формирования технологий построения крупных информационных систем. В рамках этого проекта ОИЯИ входил в состав пилотных ядерных объектов, участвующих в разработке требований к системе и реализации одной из важных задач – задачи взаимодействия Федерального информационного центра (ФИЦ) и ядерных установок. Технология построения крупных информационных систем, объединяющих большое число локальных компьютерных систем, имеет свои особенности, которые важно учитывать для успешного продвижения проекта:

- длительный жизненный цикл разработки;
- разнообразие используемого аппаратного и программного обеспечения;
- масштабность и сложность решаемых задач;
- территориальная распределенность, что особенно характерно для России и др.

Одной из условных единиц информации банка данных системы определена учетная единица. Набор параметров, описывающих учетную единицу банка данных ФИЦ, является подмножеством из описания учетной единицы, используемого в автоматизированной системе учета и контроля ЯМ ОИЯИ MTIS. Классификаторы параметров описания учетной единицы едины для ФИЦ и для информационных систем ядерных объектов, в том числе и для MTIS.

В данном проекте были решены такие вопросы, как определение интерфейса между установками и Федеральным информационным центром, организация взаимодействия между ними, форматы обменных файлов, выбор и разработка программных и технических средств подсистем, используемых для подготовки и представления отчетов в ФИС.

## Литература

1. Самойлов В.Н. Технология моделирования сложных процессов. Книга Р-10-99-173, Дубна, ОИЯИ, 1999 г. – 204 с.
2. Самойлов В.Н. Методы анализа данных физических экспериментов и сложных процессов, ОИЯИ Р10-99-105, Дубна, 1999, с. 32.
3. Ладьженский Г. Системы управления базами данных - коротко о главном. // СУБД, 1995, №2.
4. Васкевич Д. Стратегии Клиент/Сервер. - Киев: Диалектика, 1996.
5. Самойлов В.Н. «Информационная технология моделирования физических экспериментов и сложных систем», Р10-99-106, Дубна, 1999, с. 26.
6. Любарский Ю.Я. Интеллектуальные информационные системы. –М.: Наука, 1990, -227 с.
7. Мартин Дж. Планирование развития автоматизированных систем. –М.: Финансы и статистика, 1984, -196 с.
8. Системы управления базами данных и знаний. Справочное пособие. Под ред. Наумова А.М. –М.: Финансы и статистика. 1991, -241 с.
9. Васильев Г.П., Горский В.Е. Программное обеспечение неоднородных распределенных систем. –М.: Финансы и статистика, 1986. –160с.
10. Зайцев С.С. Описание и реализация протоколов сетей ЭВМ. -М.:Наука, 1989г.-272с.
11. Дубова Н. COM или CORBA? Вот в чем вопрос. Открытые системы, 1999, с. 3.
12. Thomas J. Mowbray, Phd Ron Zahavi. The Essential CORBA: System Integration Using Distributed Object, 1995.
13. Сухомлин В.М., Методологический базис открытых систем. - Открытые системы, 1996, с. 4.
14. Эккерсон В. В поисках лучшей архитектуры клиент-сервер.- Сети, 1995, с. 4
15. Robert Orfali, Dan Harkey, Jeri Edwards, The Essential Distributed Object. - John Wiley&Sons, Inc., 1996.
16. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Санкт-Петербург: Питер.2000, 642 с.
17. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.
18. Поступальский П.А. Исследование и разработка методов верификации протоколов распределённых систем на основе бисимуляционной эквивалентности сетей Петри. // Диссертация на соискание учёной степени кандидата технических наук. Владивосток 1997 г.
19. Теория АТМ и ее реализация. Учебный курс. Руслан Коммуникейшнз, 1999г.
20. Буч Г. Объектно-ориентированное проектирование с примерами применения. - М.:Конкорд, 1992.
21. Samoilov V.N., Checker A.V., Shestakov B.A., Dobrianskii V.M., Koltin G.P., Heinberg C.L., Cowley P.J., Stottlemyre A.J. «Materials Tracking Information System: A Computerized Nuclear Materials Accounting System for Tracking Items at Russian

Nuclear Facilities». 39th Annual Meeting Proceedings of the Institute of Nuclear Materials Management, Vol. XXVII, Naples, Florida, July 26-30, 1998.

22. Труды международного семинара «Учет и контроль ядерных материалов». Ангарск, 12-15 июля 1998.

23. Труды международного семинара «Разработка информационной компьютеризированной системы учета и контроля ядерных материалов России», Дубна, 16-20 мая 1999.

24. Труды Российской международной конференции по учету, контролю и физической защите ядерных материалов. Обнинск, 9-14 марта, 1997. ГНЦ РФ ФЭИ. Т. 1, Т. 2, Т. 3, 813 с.

Рукопись поступила в издательский отдел  
3 августа 2000 года.