

Системный анализ и моделирование

Тема 4. ТЕХНОЛОГИИ СИСТЕМНОГО АНАЛИЗА

Понятие технологии системного анализа

Несмотря на общепризнанную целесообразность применения системного анализа для разрешения сложных проблем, масштабы реального использования аппарата системного подхода не велики.

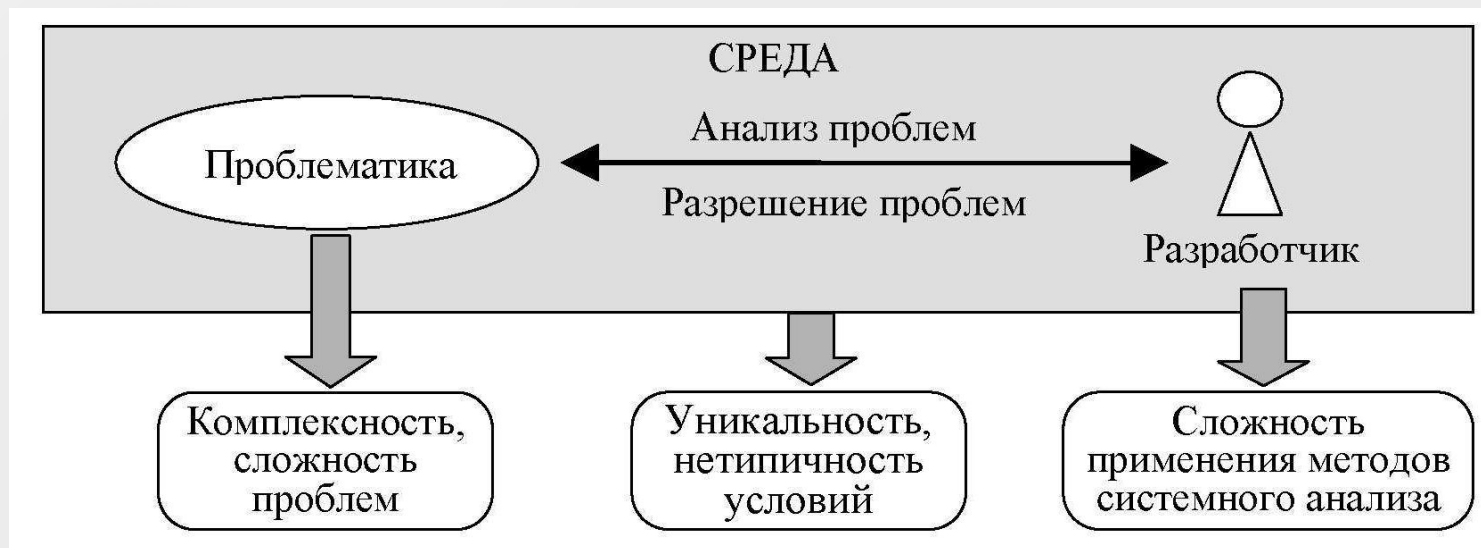
Основная причина - **сложность** самого процесса системного анализа.

Можно выделить три группы факторов сложности, связанные с тремя составляющими — **объектом системного анализа, средой и субъектом.**

Объект - исходная проблематика,

Среда - условия возникновения проблем и их разрешения,

Субъект - разработчик, осуществляющий анализ и поиск средств разрешения проблем



Факторы, связанные с объектом

Это, прежде всего, комплексность, сложность проблем.

Большинство проблем, разрешаемых с помощью системного анализа, затрагивают различные аспекты деятельности — технический, экономический, социальный, экологический и т. д.

Заметной тенденцией последнего времени является усложнение проблем, перерастание их в проблематику, т. е. в сплетение, клубок проблем, которые неразрывно связаны.

Факторы, связанные со средой

Это уникальность, нетипичность условий.

Многообразие проблем связано:

- с разнообразием условий их возникновения,
- разнообразием целей, выдвигаемых различными заинтересованными сторонами,
- имеющимися ресурсами,
- используемыми инструментальными программными средствами.

Предположение о возможности создания технологии синтеза проблеморазрешающих систем, приложимой к достаточно широкому кругу проблем, базируется на том основании, что при проектировании подобных систем, особенно на начальных стадиях, на первый план выходят общесистемные задачи.

В этих задачах основными являются **информационные, реляционные и структурные аспекты**, в то время, как **тип сущностей**, образующих систему, имеет значительно меньшее значение.

Факторы, связанные с субъектом

Это сложность применения методов системного анализа.

Трудности, которые испытывает разработчик, прежде всего, связаны со слабой формализованностью, «рыхлостью» методов системного анализа, требующих творческого осмысления и применения.

Во многом эффективность использования методов зависит от опыта и квалификации разработчика.

Системная технология

Призвана ***стандартизировать и регламентировать*** процесс выработки средств решения сложной проблемы и в конечном итоге облегчить данный процесс, уменьшить трудозатраты на его осуществление и повысить качество принимаемых решений.

Термин **технология** (от греч. *techne* — «искусство, мастерство, умение» и *logos* — «понятие, учение») широко используется в производственной сфере

Технология - последовательность согласованных технологических операций в процессе производства продукции.

Системная технология - это *информационная технология*, т. е. организованная совокупность методов обработки информации.

Использование компьютерных инструментальных систем поддержки может значительно повысить эффективность разработки за счет автоматизации рутинных операций, контроля ошибок, использования типовых фрагментов моделей, систематизации накапливаемой в процессе разработки информации, возможности проведения экспериментов на моделях и т. д.

Три основных составляющих информационной технологии:

Регламент — руководящие указания по составу этапов и их последовательности;

Методы (методы формирования моделей и поиска оптимальных решений на моделях), используемые на различных этапах системного анализа;

Инструментальные средства сопровождения.

Универсальной общесистемной технологии, предназначенная для проектирования сложных систем любой природы, пока не существует.

Наиболее развитые автоматизированные технологии:

- разработки информационных систем (CASE-технологии),
- разработки бизнес-систем (технологии реинжиниринга бизнес-процессов)
- разработки технических систем (САПР-технологии).

CASE-технологии разработки информационных систем

CASE-технологии - совокупность методологий проектирования и сопровождения информационных систем (ИС) на всем их жизненном цикле, поддержанную комплексом взаимоувязанных CASE-средств.

В основе регламента:

Жизненный цикл системы - отражает укрупненные этапы ее создания и эксплуатации.

Основные этапы жизненного цикла:

- анализ требований;
- проектирование;
- реализация (программирование);
- тестирование и отладка;
- эксплуатация и сопровождение.

Этапы, общие для большинства методик:

- формирование требований;
- разработка концепции;
- техническое задание;
- эскизный проект;
- технический проект;
- рабочая документация;
- ввод в действие;
- сопровождение.

Модели жизненного цикла

1. **Каскадная** (или **водопадная**) модель

(традиционная схема, используемая в 1970-е — начале 1980-х гг.) - предполагает строгое детерминированное следование этапов анализа, проектирования, реализации, внедрения и эксплуатации ИС по единому, заранее разработанному плану.

Анализ требований



Проектирование



Программирование



Компоновка



Тестирование



Использование

Достоинства:

- детерминированность,
- логичность,
- простота.

Основной недостаток - схемой не предусмотрена корректировка ранее принятых решений.

2. Спиральная модель (1980-х гг.) - непрерывный процесс разработки и развития ИС с планируемыми точками передачи в эксплуатацию новых версий и новых функциональных блоков.

Разработка проекта ведется как бы по спирали, на каждом витке которой создается очередная версия.

Виток спирали представляет собой законченный проектный цикл по типу каскадной схемы, т. к. предполагает последовательное выполнение всех этапов от анализа до внедрения



Достоинства:

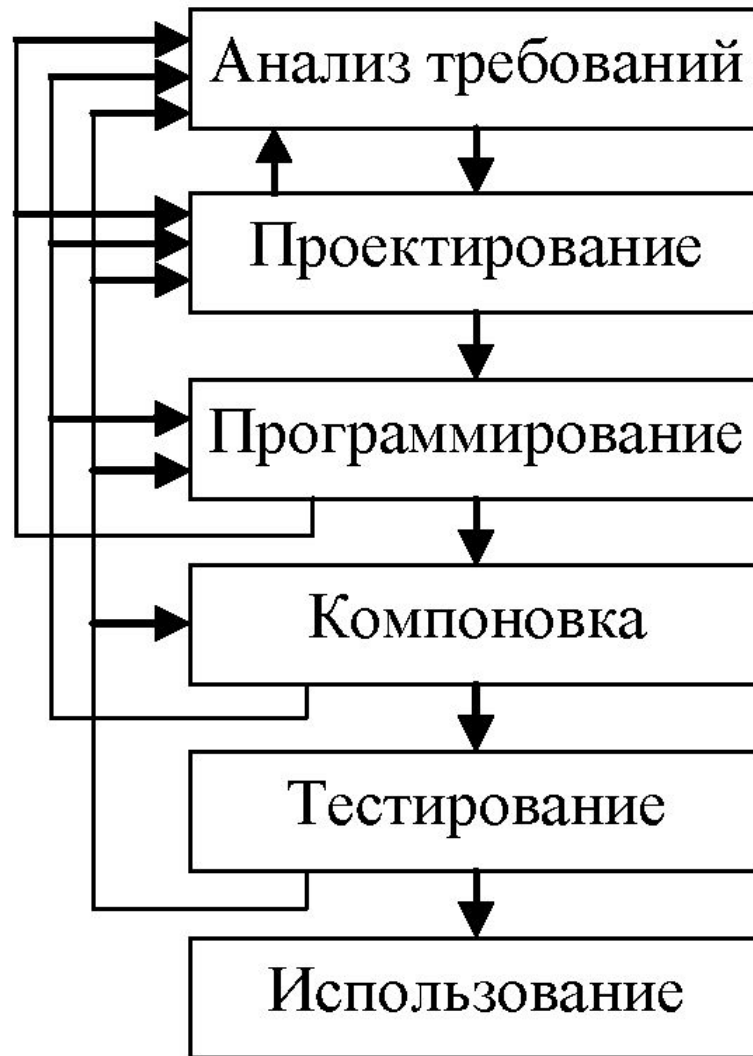
Такой подход обеспечивает на каждом витке уточнение требований к системе,

Недостатки:

Сроки разработки готового продукта по сравнению с каскадной схемой удлиняются, а затраты существенно возрастают.

3. Макетная модель (модель быстрого прототипирования (rapid prototyping)) (конец 1980-х гг.)

Последовательность этапов в данной модели внешне выглядит как каскадная, однако содержание технологических этапов таково, что многие проектные решения в процессе разработки ИС подвергаются многократным уточнениям и корректировкам, как это предусмотрено спиральной моделью.



Методы и процедуры разработки информационных систем

Первоначально создание программных продуктов сводилось к разработке алгоритмов и кодированию их в виде программных строк.

Постепенно с усложнением задач автоматизации появился целый ряд новых принципов разработки крупномасштабных систем:

- нисходящее проектирование,
- структурное программирование,
- модульное конструирование,
- сквозной структурный контроль и др.

Наибольшее распространение получили следующие методы и средства:

- SADT (Structured Analysis and Design Technique),
- DFD (Data Flow Diagrams),
- ERD (Entity-Relationship Diagrams),
- объектно-ориентированное программирование (ООП)
- средства «визуального» программирования или средства быстрой разработки приложений (RAD — Rapid Application Development)

Технологии реинжиниринга бизнес-процессов

Термин «реинжиниринг бизнес-процессов» (BPR — Business process reengineering) трактуется в настоящее время двояко.

Узкое определение ввел М. Хаммер:

Реинжиниринг бизнес-процессов - фундаментальное переосмысление и радикальное перепроектирование деловых процессов для достижения резких, скачкообразных улучшений в решающих современных показателях деятельности компании, таких как стоимость, качество, сервис и темпы.

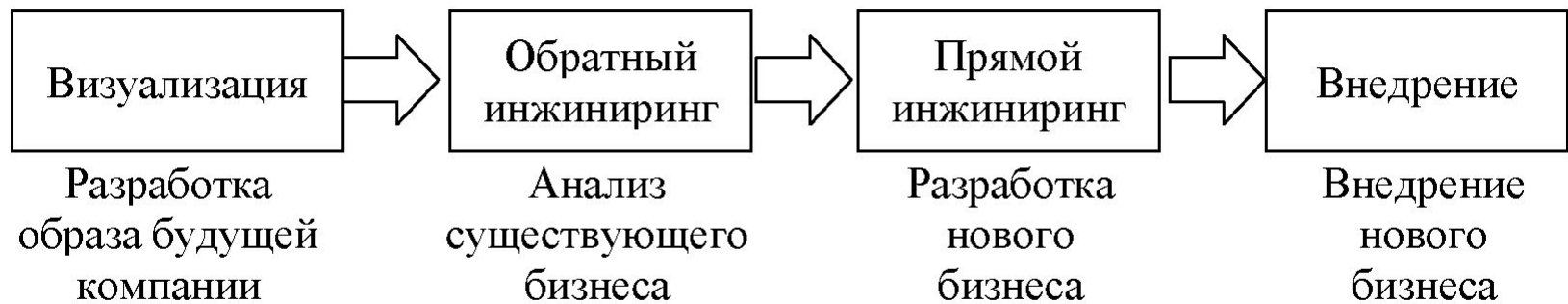
Более расширенное толкование:

Реинжиниринг - не только радикальное перепроектирование, но и постепенная реорганизация деятельности предприятия.

Этапы технология BPR:

- 1. Визуализация** — разработка образа будущей компании, создание спецификации целей реинжиниринга.
- 2. Обратный инжиниринг** — создание модели существующего бизнеса, анализ построенной модели.
- 3. Прямой инжиниринг** — разработка модели нового бизнеса и систем поддержки (системы управления, информационной системы).
- 4. Внедрение** — внедрение перепроектированных процессов.

Последовательность проведения реинжиниринга



Основным средством реинжиниринга является моделирование бизнес-процессов.

На этапе обратного инжиниринга строится модель существующего бизнеса — модель «как есть» (As is), отражающая текущее состояние системы.

На этапе прямого инжиниринга строится модель нового бизнеса — модель «как должно быть» (To be), отражающая проектируемое состояние.

Технологии проектирования технических систем

Современные технологии проектирования сложных технических объектов на основе систем автоматизированного проектирования (САПР) охватывают весь цикл разработки изделий — от выработки концепции до создания опытного образца и запуска его в производство.

Объектно-ориентированная технология системного анализа

Требования к регламенту.

- универсальность, высокая степень обобщенности, позволяющая применять его для разрешения разнообразных проблемных ситуаций;
- гибкость, простота адаптации, настройки на конкретную предметную область.

Требования к методологии моделирования.

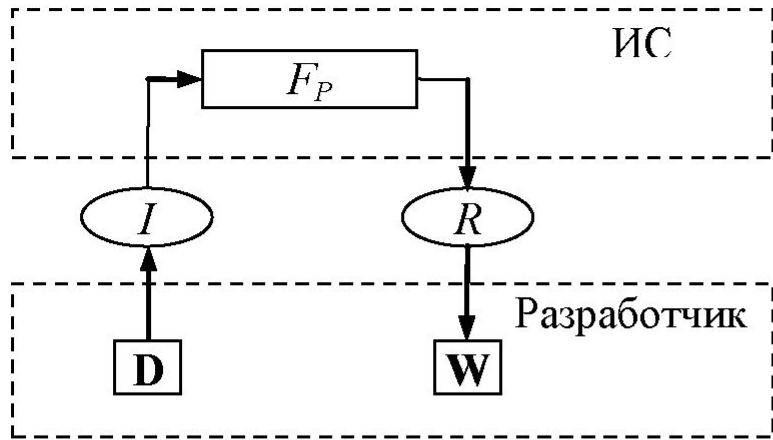
- наглядность и обозримость формируемой модели проблемосодержащей (проблеморазрешающей) системы;
- использование при построении моделей опыта экспертов.

Требования к инструментальным средствам.

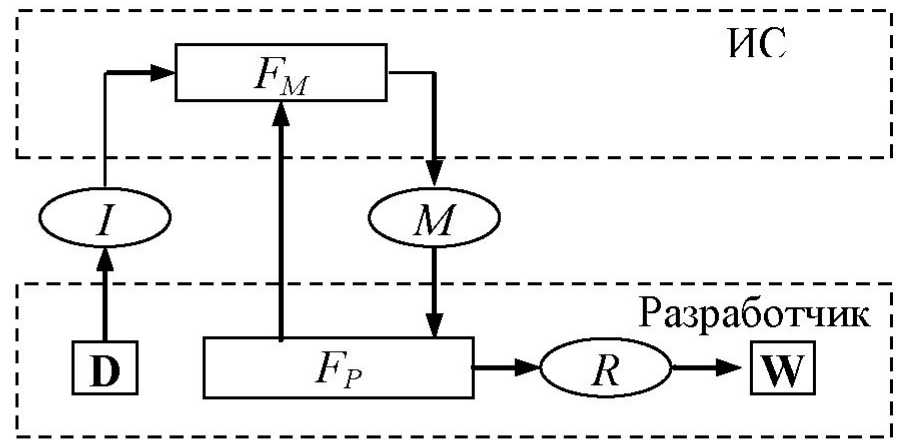
- открытость и интегрируемость;
- возможность расширения, а также сопряжения с другими приложениями, реализующими различные методики.

Принципы создания системной технологии

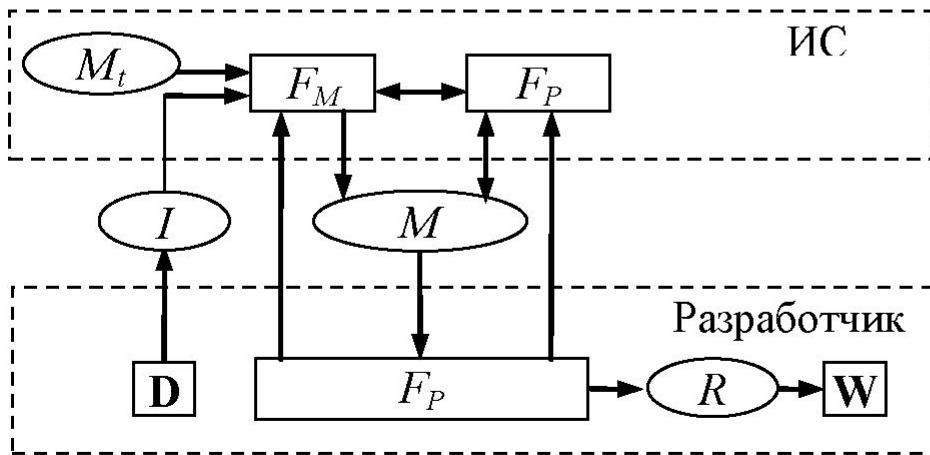
- 1. Принцип декларативности:** регламент должен предписывать вид декларативной модели, формируемой на каждой стадии, и совокупность методов, используемых для ее построения и принятия решений на модели.



a



б



в

- | | | | |
|--------------|-----------------|----------------|------------------------------|
| I (oval) | Исходные данные | D (square) | Сбор информации |
| M (oval) | Модель | W (square) | Реализация решения |
| R (oval) | Решение | F_M (square) | Процедуры построения моделей |
| M_t (oval) | Типовые модели | F_P (square) | Процедуры принятия решений |

Алгоритмическая схема (рис. *a*) предполагает, что последовательность преобразований исходной информации I в конечное описание R воплощена в некоторый единый универсальный алгоритм F , выполняемый информационной системой. Разработчику необходимо реализовать лишь процедуру сбора информации D и реализации готового проекта W .

Модельная схема (рис. б) предполагает формирование декларативной модели M проектируемой системы с помощью инструментального средства. Инструментарий содержит процедуры построения модели Fm , используемые разработчиком в интерактивном режиме для формирования модели системы «как есть» (As is) или «как должно быть» (To be). Непосредственно же принятие решений (с помощью соответствующих процедур Fp) осуществляется самим разработчиком.

Гибридная схема (рис. в) - инструментальная система, построенная по принципу систем поддержки, вместо единой процедуры содержит библиотеку процедур, которые наряду с неавтоматизированными процедурами принятия решений из общей совокупности процедур F_p могут использоваться на различных этапах разработки проблеморазрешающей системы. Декларативная модель M служит интегрирующим звеном для множества процедур. Подсистема моделирования F_m инструментального средства позволяет разработчику формировать модель, используя при этом библиотеку типовых моделей M_t и библиотеку процедур F_p .

2. Принцип иерархичности: разработка проблеморазрешающей системы должна осуществляться на разных уровнях абстрагирования с использованием иерархического подхода.

3. Принцип итеративности: схема применения этапов системного анализа должна быть итеративной.

4. Принцип типизации: при поиске решений для устранения сложной проблемы следует использовать типовые знания.

5. Принцип комплексирuемости: должна быть предусмотрена возможность сочетания, интегрирования разнообразных методов и процедур принятия решений и соответствующих программных компонент.

Для реализации принципа комплексирuемости целесообразно использовать объектно-ориентированный подход.

Однако существующие языки ООАП, в частности UML, не совсем приемлемы для принятия решения при анализе и проектировании сложных систем различного назначения:

1. Они в основном предназначены для проектирования информационных систем, в связи с чем оперируют собственным понятийным аппаратом, ориентированным на данную область.
2. ООАП-методы и средства используются лишь для декларативного описания структуры и свойств системы. Как правило, они не содержат процедур генерации и выбора вариантов реализации систем.

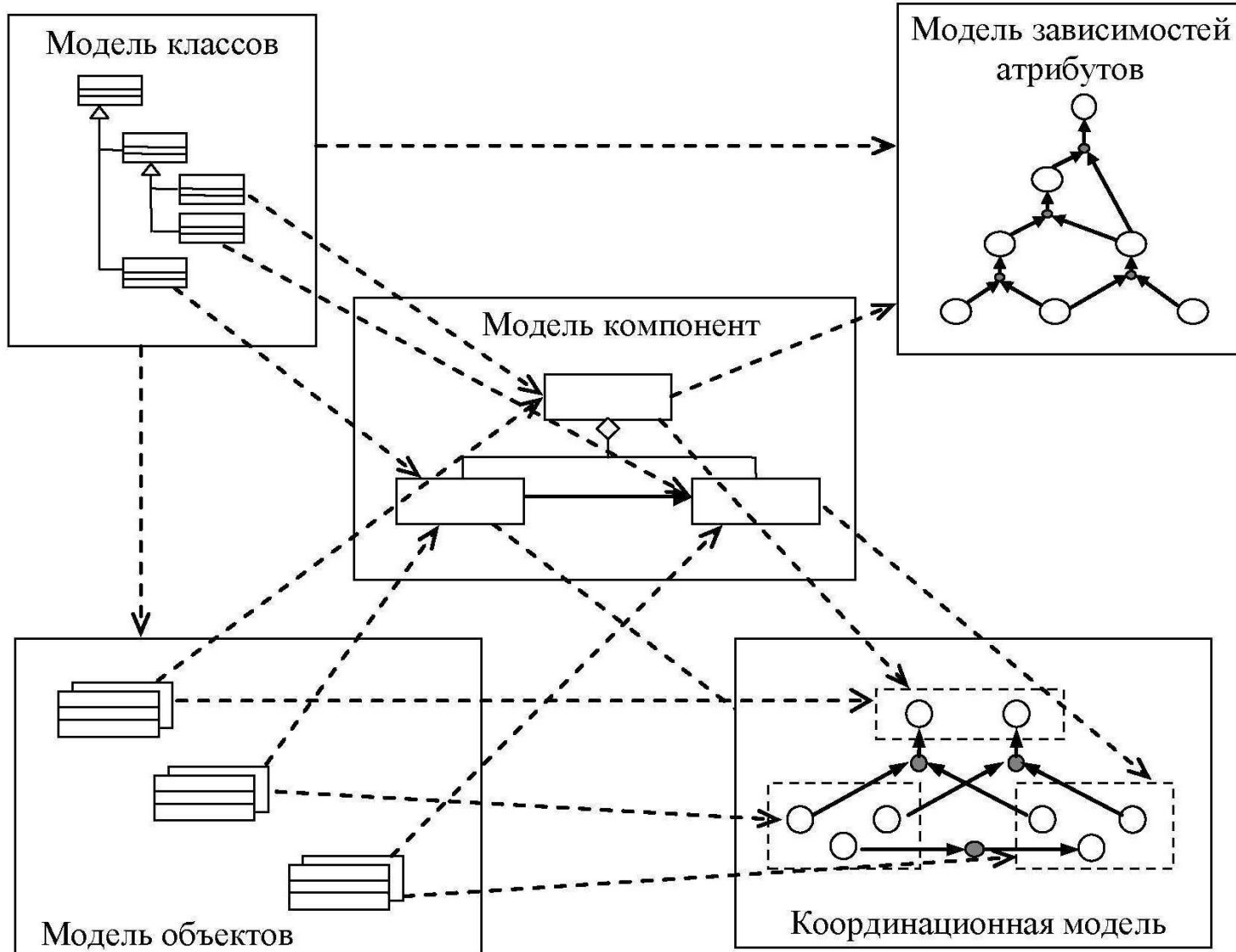
Объектно-ориентированная методология моделирования OMSD

Разработана специально для целей анализа и синтеза сложных систем различной природы.

Модель сложной системы может включать пять видов моделей:

- компонент,
- классов,
- объектов,
- зависимостей атрибутов,
- Координационную.

Составляющие модели системы



Каркасом является модель компонент, отражающая структуру системы.

Для каждой из компонент на основе модели классов может быть создана модель объектов, а также модель зависимостей атрибутов.

Модель объектов содержит различные варианты реализации компоненты.

Модель зависимостей атрибутов помогает определять варианты и осуществлять поиск оптимальных.

Для согласования вариантов реализации компонент, представляемых как объекты, формируются координационные модели.

Модель компонент

Компонент - любая часть предметной области, которая может быть выделена и описана как некоторая самостоятельная сущность.

Два вида компонент — **подсистема** (в том числе и система в целом) и **элемент**.

Подсистемы выделяются путем последовательной декомпозиции системы.

Стратифицированная иерархия («Part-of-иерархия») - совокупность подсистем, полученных в результате декомпозиции одной и той же системы предыдущего уровня, подуровнем.

Порождающая подсистема - материнская, а подсистемы, составляющие подуровень — дочерние.

Между материнской системой и ее дочерними подсистемами устанавливается *отношение агрегации* («целое - часть»).

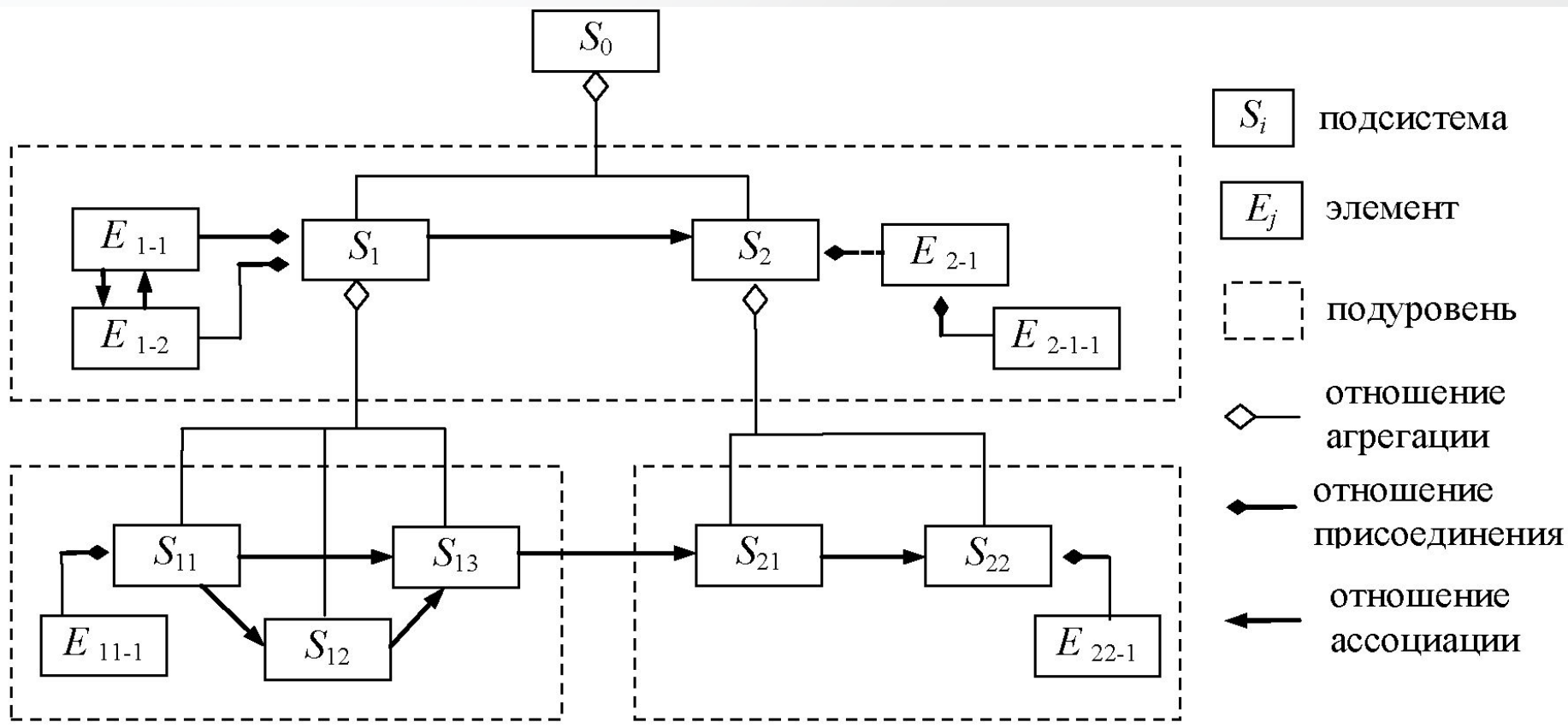
Формирование любого подуровня может осуществляться с помощью стандартных оснований декомпозиции.

Кроме подсистем могут быть выделены **элементы** — сущности, участвующие в деятельности подсистемы (например: конечные продукты, предметы деятельности, средства деятельности, исполнители), или комплексные свойства, описываемые набором характеристик (например: технологические параметры, экономические результаты, технические условия и т. д.).

Элементы «присоединяются» к подсистеме посредством *отношения присоединения*.

Разнообразные отношения, отражающие физические связи (*отношениями ассоциации*) - потоки, заключающиеся в передаче энергии, вещества или информации, а также разнообразные эмпирические отношения (например: «руководить», «владеть», «использовать»).

Структура взаимосвязи компонент сложной системы



Модель классов

Чтобы для каждого компонента (подсистемы или элемента) из иерархии компонент можно было осуществлять поиск решения (оптимального варианта реализации данного компонента), необходимо составить структурированное описание компонент.

Для описания наиболее удобен **объектный подход**, предполагающий описание в виде совокупности *атрибутов*, отражающих свойства, характеристики компонента, и совокупности *методов*, позволяющих манипулировать атрибутами (задавать их значения, вычислять).

Структура описания задается в виде класса.

Класс - структура, состоящая из имени класса, множества атрибутов и множества методов (присоединенных процедур).

Для каждого атрибута задается *имя, тип атрибута* (<String>, <Real>, <Integer>, <Date>, <Object>, ...) и множество допустимых значений (домен) атрибута. Домен задается в зависимости от типа атрибута. Так, для атрибута типа <Real> или <Integer> — в виде интервала, для атрибута типа <String> — в виде списка строк, для атрибута типа <Object> — в виде имени класса (списка имен классов)

Примеры классов и объектов



Иерархия типовых классов, отражающих общие свойства для различных классов компонент систем, может быть составлена на основе знаний экспертов и сохранена в библиотеке типовых моделей.

Разработчик может использовать эти знания - описание для конкретной подсистемы в этом случае может создаваться путем выбора или наследования подходящего класса из библиотеки классов и заполнения его конкретными данными (значениями атрибутов).

Использование типовых классов существенно облегчает процесс построения декларативной модели.

Модель объектов

На базе классов создаются *объекты* (экземпляры классов), содержащие конкретные значения атрибутов. При присвоении значений атрибутам осуществляется проверка на принадлежность присваиваемого значения области допустимых значений (домену).

Значение атрибута может быть дополнено *коэффициентом уверенности*, представляющим собой число в интервале от 0 до 1 (1 соответствует абсолютной уверенности в значении атрибута, 0 — абсолютной неуверенности, т. е. незнанию).

Для отображения множества вариантов реализации некоторого класса, а также множества состояний системы (подсистемы, элемента) в различные моменты времени, в различных точках пространства, в различных условиях используется так называемый *мультиобъект*.

Мультиобъект - множество объектов (экземпляров) одного класса, выделенных в соответствии с некоторым *признаком*.

Признак - специальный атрибут, принимающий дискретные значения.

Три основных типа базовых признаков:

- время,
- пространство,
- группа.

От них могут наследоваться конкретные признаки. Признак может быть *комбинированным*, т. е. являться композицией нескольких признаков.

Каждому значению признака соответствует некоторый конкретный экземпляр.

На схемах мультиобъекты изображаются и в виде стопки объектов (в соответствии с обозначениями, принятыми в языке UML), и в табличном виде.

Пример мультиобъекта

<u>Производство коробок (3):</u>
<u>Производство коробок (2):</u>
<u>Производство коробок (1):</u> <u>Текущее производство продукта</u>
<i>Период, месяц: 1</i> Продукт: <u>Коробка</u> Объем производства: 300 Численность персонала: 3 Производительность: 100 Затраты, тыс. р.: 60 Затраты на ед. продукции: 0,2 Эффективность: ср / 0.8

<u>Производство</u> <u>коробок</u>	<i>Период, месяц</i>		
	1	2	3
Объем производства	300	240	270
Численность персонала	3	3	3
Производительность	100	80	90
Затраты, тыс.р.	60	65	60
Затраты на ед. продукции	0,2	0,27	0,22
Эффективность	выс/1	ср/0.8	ср/0.6

Мультиобъекты могут использоваться для:

- *ретроспективного* или *сравнительного* анализа,
- формирования альтернативных вариантов реализации компонент системы, их оценки и выбора оптимальных вариантов

Модель зависимостей атрибутов

Атрибуты могут зависеть друг от друга: значения одних могут вычисляться через значения других.

Модель зависимостей атрибутов создается на базе некоторого класса, сопоставленного некоторой подсистеме.

Атрибуты делятся на:

Независимые (базовые) атрибуты - описывают первичные, непосредственные характеристики подсистемы (например: «управляемые», «конструктивные» параметры, «внешние возмущения», например, параметры, определяющие принципы работы подсистемы, характеристики условий функционирования, данные наблюдений (мониторинга) и т. д.).

Целевые атрибуты - определяют целевое состояние подсистемы (например: показатели эффективности и качества подсистемы (ее деятельности)).

Значения базовых атрибутов разработчик может задавать непосредственно.

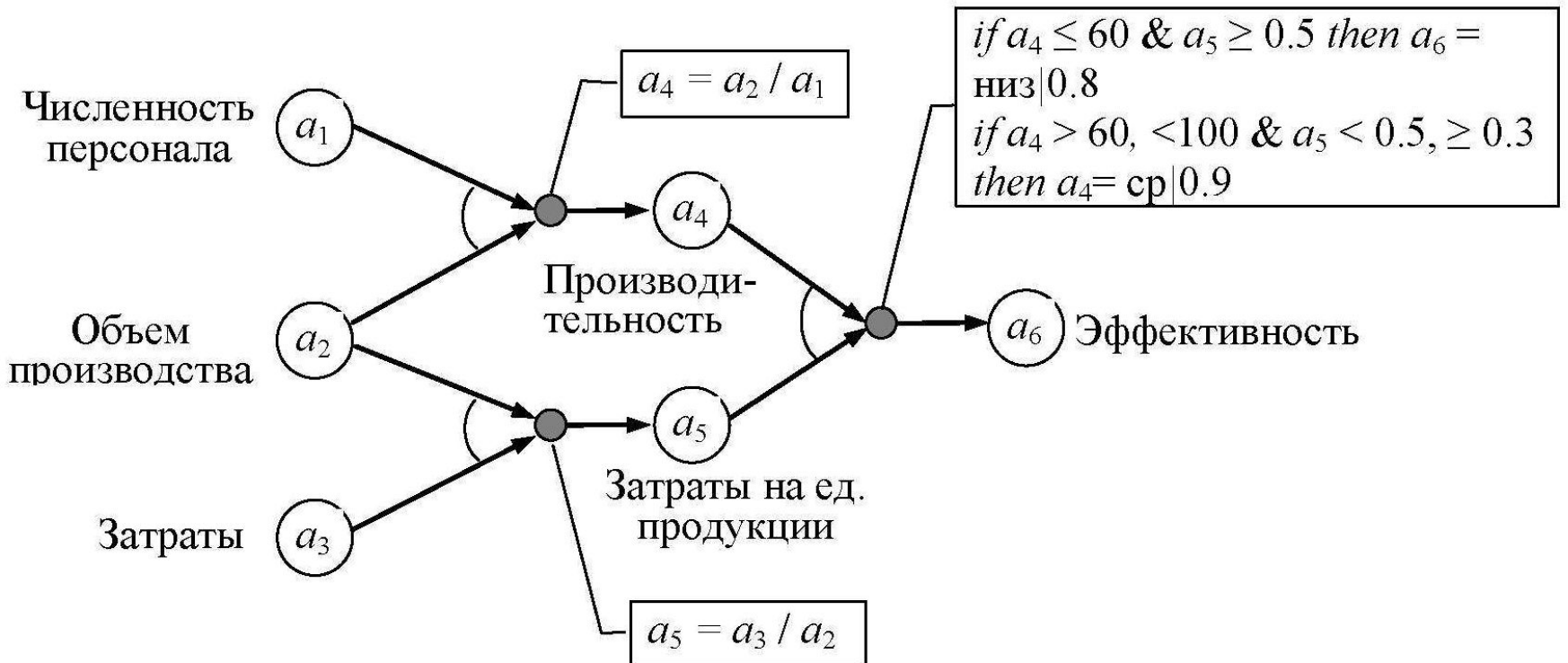
Значения целевых атрибутов зависят от значений базовых атрибутов, являются их следствием.

Влияние базовых атрибутов на целевые (через промежуточные) задается с помощью *отношений функциональной зависимости*.

Каждое такое отношение связывает атрибут-функцию с атрибутами-аргументами.

Множество отношений зависимости представляют собой сеть, в истоках которой находятся базовые атрибуты, в стоках — целевые атрибуты.

Сеть функциональных зависимостей атрибутов



Закономерность, показывающая, как именно значение атрибута-функции определяется значениями атрибутов-аргументов, может задаваться в виде формулы, некоторой процедуры или совокупности правил-продукций формата if ... then (если ... то).

Причем правила могут характеризоваться степенью уверенности. Условная часть правила содержит логические формулы для проверки значений атрибутов-аргументов, заключительная — значение атрибута-функции (либо формулу для вычисления, либо процедуру).

Модель функциональных зависимостей атрибутов используется для решения двух основных классов задач:

- 1. Задача интерпретации (прямая задача).** Исходными данными являются значения базовых атрибутов. Необходимо найти значения некоторых целевых атрибутов, соответствующих заданным исходным данным. К данному типу относятся задачи диагностики, оценки, прогнозирования. Для решения прямой задачи используются методы прямого вывода.
- 2. Задача поиска допустимого решения (обратная задача).** Исходными данными является определенное (заданное) значение целевого атрибута. Необходимо найти соответствующие значения базовых атрибутов. К данному типу относятся задачи планирования, целевого управления. Для решения обратной задачи используются методы обратного вывода.

Координационная модель

Используется для отражения отношения зависимости между атрибутами классов, сопоставленных различным компонентам системы.

Два типа отношений зависимостей, включаемых в модель координации:

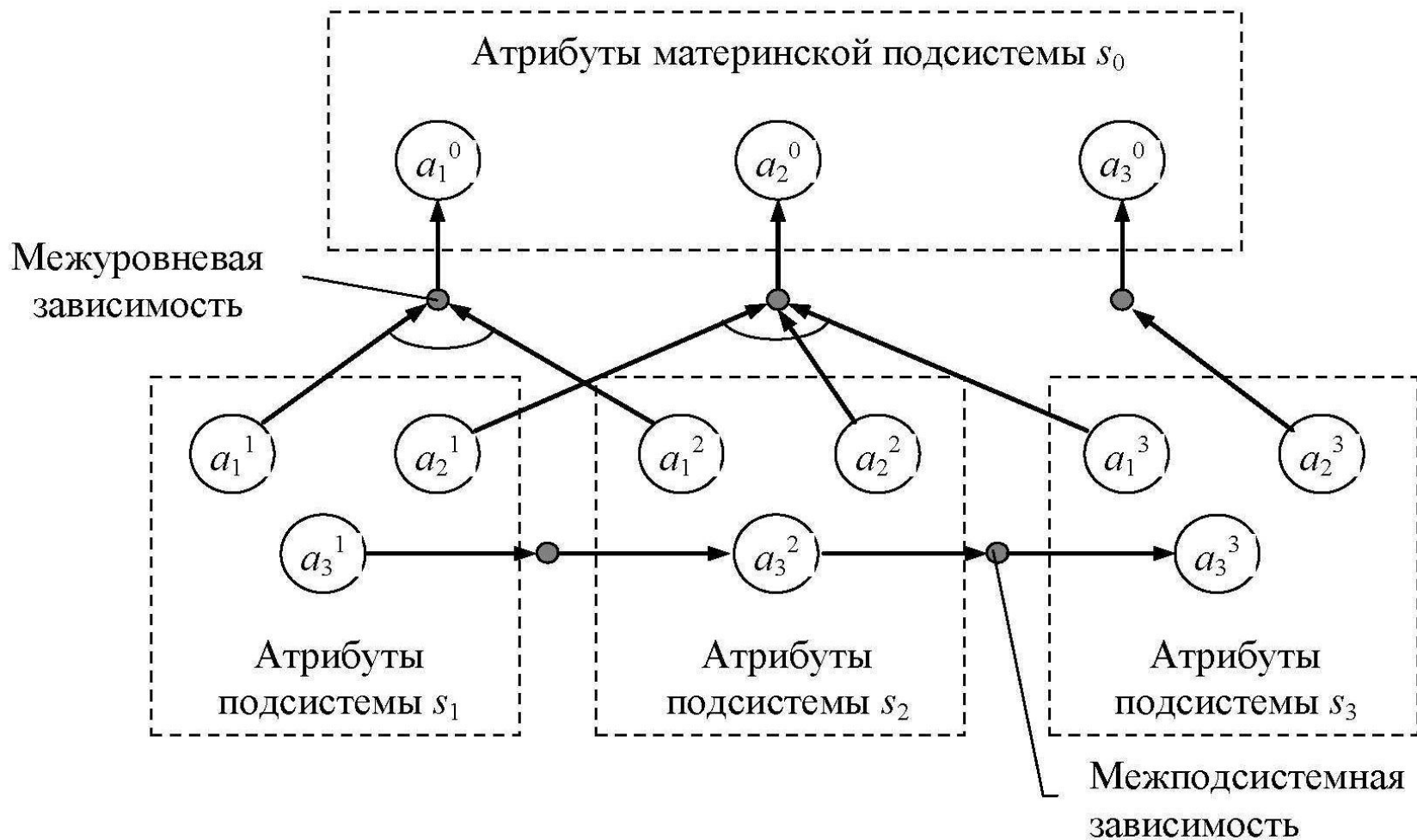
1. **Межуровневые зависимости** - связывают атрибуты материнской подсистемы с атрибутами дочерних подсистем (или элементов).

Пример: зависимость затрат материнской системы от затрат отдельных подсистем, определяемая как сумма.

2. **Межкомпонентные зависимости** - это либо зависимости атрибутов подсистем, входящих в подуровень (межподсистемные зависимости — баланс входов-выходов подсистем), либо зависимости атрибутов элементов одной и той же подсистемы (межэлементные зависимости).

Пример межподсистемных зависимостей: объем продукции, производимой некоторой подсистемой, должен быть равен объему продукции, поступающей на вход другой подсистемы, умноженному на коэффициент потерь, возникающих при передаче.

Координационная модель подуровня



Координационная модель используется для согласования вариантов (реализаций) подсистем подуровня друг с другом и с вариантом материнской подсистемы, а также для согласования вариантов элементов в рамках подсистемы.

Два класса задач, решаемых с помощью координационной модели:

- 1. Задача интеграции.** На основе известных значений атрибутов дочерних компонент определяются значения атрибутов материнской подсистемы (интегрированных показателей) с использованием межуровневых зависимостей. При этом значения атрибутов дочерних компонент, зависящие от значений атрибутов других компонент, могут определяться с использованием межкомпонентных зависимостей.
- 2. Задача координации.** Значения атрибутов материнской подсистемы выступают ограничениями при поиске допустимых значений атрибутов дочерних компонент. Кроме того, одни дочерние компоненты могут накладывать ограничения на другие. Ограничения задаются на основе межуровневых и межкомпонентных зависимостей.

Регламент объектно-ориентированной технологии

Представляет собой руководящие указания по составу этапов и их последовательности, а также используемым на каждом из этапов методам формирования моделей и поиска решений.

В соответствии с принципом декларативности регламент представляется не в виде алгоритма, а в виде совокупности рекомендаций для каждого из основных этапов типовой последовательности относительно содержания этапа, вида формируемых моделей и спектра методов, которые могут быть использованы разработчиком для формирования модели и поиска решения на ней.

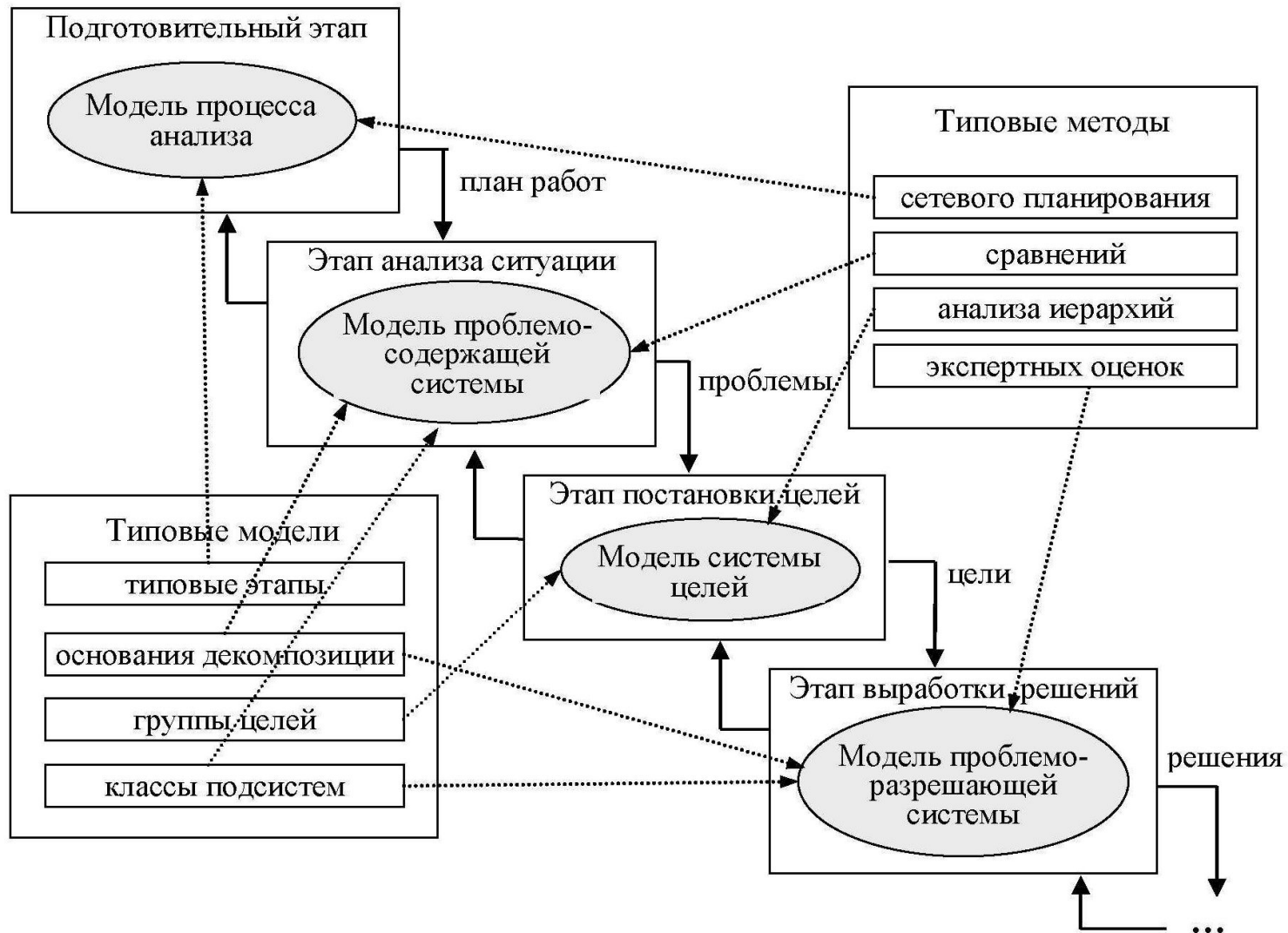
Подготовительный этап - формируется модель процесса проведения системного анализа.

Этап анализа — модель проблемосодержащей системы,

Этап постановки целей — модель системы целей,

Этап выработки решений — модель проблеморазрешающей системы

Схема выполнения этапов системной технологии



Подготовительный этап

Выбор способа организации процесса проведения системного анализа, в том числе: планирование (определение последовательности этапов, описание работ, исполнителей, используемых методов и средств), а также создание коллектива разработчиков.

Этап анализа ситуации

Исследуется проблемосодержащая система — выявляются ее сильные и слабые стороны, тенденции, проблемы.

В зависимости от целей анализа проводится один или несколько различных его видов — ретроспективный, сравнительный, причинный анализ.

Предварительным этапом при использовании любого вида анализа является построение декларативной модели в виде иерархии компонент, дополненной структурированными описаниями компонент и, возможно, моделью зависимостей атрибутов.

Этап постановки целей

Формулируются цели и задачи исходя из проблем, выявленных на этапе анализа ситуации.

Формирование модели системы целей начинается с построения иерархии компонент.

В качестве подсистем здесь выступают группы целей, выделенные, например, в соответствии с различными аспектами целеполагания — политическим, экономическим, социальным, экологическим.

Цели могут быть сгруппированы и в соответствии с подсистемами внешней среды, выдвигающими требования к исследуемой системе, — цели вышестоящих систем, потребителей, поставщиков, партнеров и т. д.

Этап выработки решений

Выработка решений для ликвидации проблем, выявленных на этапе анализа, исходя из целей и задач, сформулированных на этапе постановки целей.

Разработка осуществляется иерархически. На верхнем уровне осуществляется выработка обобщенного решения, затем решение уточняется на более низких уровнях. Таким образом, сначала формируется иерархия подсистем и элементов.

Иерархия может соответствовать иерархии компонент проблемосодержащей системы, построенной на этапе анализа, или может быть построена заново, например, в соответствии с целями и задачами, выявленными на этапе постановки целей.

Этап организации выполнения

Осуществляется разработка плана реализации принятых на предыдущем этапе решений, разработка обеспечивающих комплексов, оценка рисков.

Этап оценивания результатов

Выполняется аналогично этапу анализа ситуации.

Осуществляется сравнительный анализ состояния, достигнутого в результате реализации решений, или прогнозируемых состояний с целевым состоянием.

Для оценки используются критерии, выделенные на этапе постановки целей.