

**Модель** – описание системы, отражающее определенную группу ее свойств.

Описание системы ведется с трех точек зрения:

**функциональной, морфологической и информационной**

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

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

Система может быть однофункциональной и многофункциональной.

Оценка функций системы (в абсолютном смысле) зависит от точки зрения того, кто ее оценивает (или системы, ее оценивающей).

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

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

Функциональная организация может быть описана:

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

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

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

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

В самом общем виде функциональное описание системы в любой динамической системе изображается семеркой:

$$Sf = \{T, x, C, Q, y, \varphi, \eta\},$$

где  $T$  - множество моментов времени,

$x$  - множество мгновенных значений входных воздействий,

$C = \{c: T \rightarrow x\}$  – множество допустимых входных воздействий;

$Q$  – множество состояний;

$y$  - множество значений выходных величин;

$Y = \{u: T \rightarrow y\}$  - множество выходных величин;

$\varphi = \{T \times T \times T \times C \rightarrow Q\}$  - переходная функция состояния;

$\eta: T \times Q \rightarrow y$  – выходное отображение;

$c$  - отрезок входного воздействия;

$u$  – отрезок выходной величины.

Такое описание системы охватывает широкий диапазон свойств. Недостаток данного описания – не конструктивность: трудность интерпретации и практического применения. Функциональное описание должно отражать такие характеристики сложных и слабо познанных систем как параметры, процессы, иерархию.

Примем, что система  $S$  выполняет  $N$  функций  $\psi_1, \psi_2, \dots, \psi_s, \dots, \psi_N$ , зависящих от  $n$  процессов  $F_1, F_2, \dots, F_i, \dots, F_n$ . Эффективность выполнения  $s$ -й функции  $\mathcal{E}_s = \mathcal{E}_s(\psi_s) = \mathcal{E}(F_1, F_2, \dots, F_i, \dots, F_n) = \mathcal{E}_s(\{F_i\}), i=1 \dots n, s=1 \dots N$ .

Общая эффективность системы есть вектор-функционал  $\mathcal{E} = \{\mathcal{E}_s\}$ . Эффективность системы зависит от огромного количества внутренних и внешних факторов.

Рациональный путь формирования функционального описания состоит в применении такой многоуровневой иерархии описаний, при которой описание более высокого уровня будет зависеть от обобщенных и факторизованных переменных низшего уровня.

Иерархия создается по уровневой факторизацией процессов  $\{ Fi \}$  при помощи обобщенных параметров  $\{ Qi \}$ , являющихся функционалами  $\{ Fi \}$ . Предполагается, что число параметров значительно меньше числа переменных, от которых зависят процессы. Такой способ описания позволяет построить мост между свойствами взаимодействующих со средой элементов (подсистемами низшего уровня) и эффективностью системы.

Процессы  $\{ Fi(1) \}$  можно обнаружить на выходе системы. Это процессы взаимодействия со средой. Будем называть их процессами первого уровня и полагать, что они определяются:

- а) параметрами системы первого уровня -  $Q1(1), Q2(1), \dots, Qj(1), \dots, Qm(1)$  ;
- б) активными противодействующими параметрами среды, непосредственно направленными против системы для снижения ее эффективности -  $b1, b2, \dots, bk, \dots, bK$  ;
- в) нейтральными (случайными параметрами среды)  $c1, c2, \dots, cl, \dots, cL$  ;
- г) благоприятными параметрами среды  $d1, d2, \dots, dp, \dots, dP$  .

Среда имеет непосредственный контакт с подсистемами низших уровней, воздействуя через них на подсистемы более высокого уровня иерархии, так что  $Fi^* = Fi^* (\{bk\}, \{cl\}, \{dp\})$  .

Путем построения иерархии (параметры  $b$ -го уровня – процессы  $(b-1)$ -го уровня – параметры  $(b-1)$ -го уровня) можно связать свойства среды с эффективностью системы. Параметры системы  $\{Q_j\}$  могут изменяться при изменении среды, они зависят от процессов в системе и записываются в виде функционалов состояния  $Q_{j1}(t)$ .

Собственным функциональным пространством системы  $W$  называется пространство, точками которого являются все возможные состояния системы, определяемое множеством параметров до уровня  $b$ :  $Q = \{Q(1), Q(2), \dots, Q(b)\}$ . Состояние может сохраняться постоянным на некотором интервале времени  $T$ . Процессы  $\{F_i(2)\}$  не могут быть обнаружены на выходе системы. Это процессы второго уровня, зависящие от параметров  $Q(2)$  подсистем системы (параметров второго уровня). И так далее.

Образуется иерархия описания: эффективность (конечное множество функционалов) – процессы первого уровня (функции) – параметры первого уровня (функционалы) – процессы второго уровня (функции) – параметры второго уровня (функционалы) и т.д. На каком-то уровне наши знания о функциональных свойствах системы исчерпываются, и иерархия обрывается. Обрыв может произойти на разном уровне для разных параметров (процессов), причем как на процессе, так и на параметре. Внешние характеристики системы определяются верхним уровнем иерархии, поэтому часто удается ограничиться описанием вида  $(\{E_i\}, \{\psi_S\}, \{F_i(1)\}, \{Q_j(1)\}, \{b_k\}, \{c_l\}, \{d_p\})$ . Число уровней иерархии зависит от требуемой точности представления входных процессов.

## ГРАФИЧЕСКИЕ СПОСОБЫ ФУНКЦИОНАЛЬНОГО ОПИСАНИЯ СИСТЕМ

Выше был рассмотрен способ обобщенного аналитического функционального описания систем. Очень часто при анализе и синтезе систем используется графическое описание, разновидностями которого являются: дерево функций системы, стандарт функционального моделирования IDEF0.

Все функции, реализуемые сложной системой, могут быть условно разделены на три группы:

- целевая функция;
- базисные функции системы;
- дополнительные функции системы.

Целевая функция системы соответствует ее основному функциональному назначению, т.е. целевая (главная) функция – отражает назначение, сущность и смысл существования системы.

Основные функции отражают ориентацию системы и представляют собой совокупность макрофункций, реализуемых системой. Эти функции обуславливают существование системы определенного класса. Основные функции – обеспечивают условия выполнения целевой функции (прием, передача, приобретение, хранение, выдача).

Дополнительные (сервисные) функции расширяют функциональные возможности системы, сферу их применения и способствуют улучшению показателей качества системы. Дополнительные функции – обеспечивают условия выполнения основных функций (соединение (разведение, направление, гарантирование)).

Дерево функций системы представляет декомпозицию функций системы и формируется с целью детального исследования функциональных возможностей системы и анализа совокупности функций, реализуемых на различных уровнях иерархии системы. На базе дерева функций системы осуществляется формирование структуры системы на основе функциональных модулей. В дальнейшем структура на основе таких модулей покрывается конструктивными модулями (для технических систем) или организационными модулями (для организационно-технических систем). Таким образом, этап формирования дерева функций является одним из наиболее ответственных не только при анализе, но и при синтезе структуры системы. Ошибки на этом этапе приводят к созданию "систем-инвалидов", не способных к полной функциональной адаптации с другими системами, пользователем и окружающей средой.

Исходными данными для формирования дерева функций являются основные и дополнительные функции системы. Формирование дерева функций представляет процесс декомпозиции целевой функции и множества основных и дополнительных функций на более элементарные функции, реализуемые на последующих уровнях декомпозиции. При этом каждая из функций конкретно взятого  $i$ -ого уровня может рассматриваться как макрофункция по отношению к реализующим ее функциям на  $(i+1)$ -го уровня, и как элементарная функция по отношению к соответствующей функции верхнего  $(i-1)$ -го уровня.

Моделирование сложных систем (такими являются современные промышленные системы) было начато в программе интегрированной автоматизации производства (ICAM – (Integrated Computer Aided Manufacturing) Министерства обороны США в которой была признана полезность методологии SADT (Structured Analysis and Design Technique – Технология структурного анализа и проектирования) введенной в 1973 году Россом, что привело к стандартизации и публикации ее части, называемой IDEF0 (IDEF=ICAM DEFinition или *Integration Definition for Function Modeling*). С 1981 года стандарт IDEF0 претерпел несколько незначительных изменений, в основном ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологичам США (NIST). В 2000 году – IDEF0 был принят в качестве стандарта в Российской Федерации.



При этом каждая из функций конкретно взятого  $i$ -ого уровня может рассматриваться как макрофункция по отношению к реализующим ее функциям на  $(i+1)$ -го уровня, и как элементарная функция по отношению к соответствующей функции верхнего  $(i-1)$ -го уровня.

### КРАТКОЕ ОПИСАНИЕ МЕТОДОЛОГИИ IDEF0

Моделирование сложных систем (такими являются современные промышленные системы) было начато в программе интегрированной автоматизации производства (ICAM - (Integrated Computer Aided Manufacturing) Министерства обороны США в которой была признана полезность методологии SADT (Structured Analysis and Design Technique - Технология структурного анализа и проектирования) введенной в 1973 году Россом, что привело к стандартизации и публикации ее части, называемой IDEF0 (IDEF=ICAM DEFINITION или Integration Definition for Function Modeling). С 1981 года стандарт IDEF0 претерпел несколько незначительных изменений, в основном ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологиям США (NIST). В 2000 году - IDEF0 был принят в качестве стандарта в Российской Федерации.

Объектами моделирования являются системы.

Описание IDEF0 модели построено в виде иерархической пирамиды, в вершине которой представляется самое общее описание системы, а основание представляет собой множество более детальных описаний.

IDEF0 методология построена на следующих принципах:

- Графическое описание моделируемых процессов. Графический язык Блоков и Дуг IDEF0 Диаграмм отображает операции или функции в виде Блоков, а взаимодействие между входами/выходами операций, входящими в Блок или выходящими из него, Дугами.
- Лаконичность. За счет использования графического языка описания процессов достигается с одной стороны точность описания, а с другой – краткость.

Необходимость соблюдения правил и точность передачи информации. При IDEF0 моделировании необходимо придерживаться следующих правил:

- На Диаграмме должно быть не менее 3-х и не более 6-и функциональных Блоков.
- Диаграммы должны отображать информацию, не выходящую за рамки контекста, определенного целью и точкой зрения.

- Диаграммы должны иметь связанный интерфейс, когда номера Блоков, Дуги и ISOM коды имеют единую структуру.
- Уникальность имен функций Блоков и наименований Дуг.
- Четкое определение роли данных и разделение входов и управлений.
- Замечания для Дуг и имена функций Блоков должны быть краткими и лаконичными.
- Для каждого функционального Блока необходима как минимум одна управляющая Дуга.
- Модель всегда строится с определенной целью и с позиций конкретной точки зрения.

В процессе моделирования очень важным является четко определить направление разработки модели – ее контекст, точку зрения и цель.

**Контекст модели** очерчивает границы моделируемой системы и описывает ее взаимосвязи с внешней средой.

**Точка зрения** определяет позицию автора, т.е. что будет рассматриваться и под каким углом зрения.

Необходимо помнить, что одна модель представляет одну точку зрения. Для моделирования системы с нескольких точек зрения используется несколько моделей.

**Цель** отражает причину создания модели и определяет ее назначение. При этом, все взаимодействия в модели рассматриваются именно с точки зрения достижения поставленной цели.

В рамках методологии IDEF0 модель системы описывается при помощи Графических IDEF0 Диаграмм и уточняется за счет использования FEO, Текстовых и Диаграмм Глоссария. При этом модель включает в себя серию взаимосвязанных Диаграмм, разделяющих сложную систему на составные части. Диаграммы более высокого уровня (A-0, A0) – являются наиболее общим описанием системы, представленным в виде отдельных Блоков. Декомпозиция этих Блоков позволяет достигать требуемого уровня детализации описания системы.

Разработка IDEF0 Диаграмм начинается с построения самого верхнего уровня иерархии (A-0) – одного Блока и интерфейсных Дуг, описывающих внешние связи рассматриваемой системы. Имя функции, записываемое в Блоке 0, является **целевой функцией системы** с принятой точки зрения и цели построения модели.

При дальнейшем моделировании Блок 0 декомпозируется на Диаграмме A0, где целевая функция уточняется с помощью нескольких Блоков, взаимодействие между которыми описывается с помощью Дуг. В свою очередь, функциональные Блоки на Диаграмме A0 могут быть также декомпоziрованы для более детального представления.

В результате, имена функциональных Блоков и интерфейсные Дуги, описывающие взаимодействие всех Блоков, представленных на Диаграммах, образуют иерархическую взаимосогласованную модель.

Хотя вершиной модели является Диаграмма уровня А-0, настоящей "рабочей вершиной или структурой" является Диаграмма А0, поскольку она является уточненным выражением точки зрения модели. Ее содержание показывает, что будет рассматриваться в дальнейшем, ограничивая последующие уровни в рамках цели проекта. Нижние уровни уточняют содержание функциональных Блоков, детализируя их, но, не расширяя границ модели.

### ОПИСАНИЕ СИНТАКСИСА ЯЗЫКА МОДЕЛИРОВАНИЯ

Основными элементами на IDEF0 Диаграммах являются Блоки и Дуги. Блоки служат для отображения функций (действий), выполняемых моделируемой системой. Сформулированные функции должны содержать глагольный оборот.

*глагол + объект действия + [дополнение].*

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

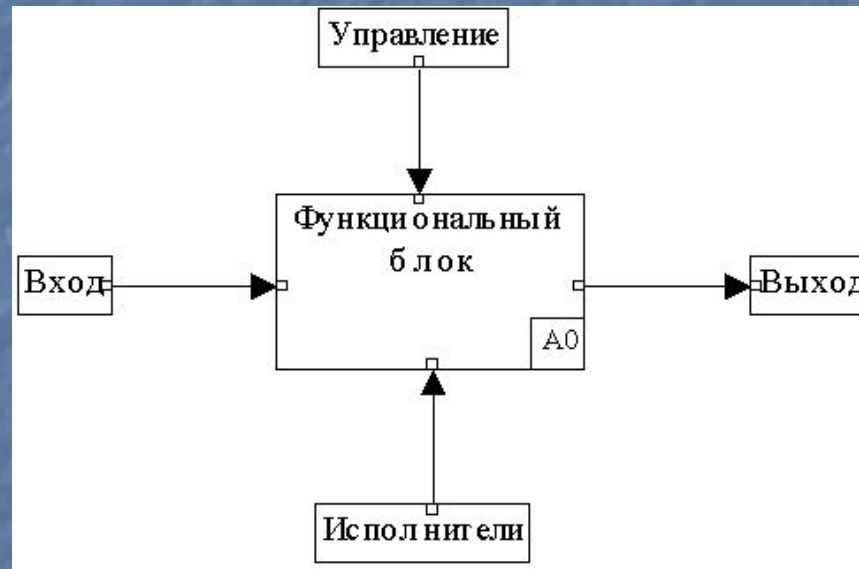
Дуги служат для отображения информации или материальных объектов, которые необходимы для выполнения функции или появляются в результате ее выполнения (объекты, обрабатываемые системой). Под объектами в рамках функционального моделирования могут пониматься документы, физические материалы, инструменты,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

Место соединения дуги с блоком определяет тип интерфейса.

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

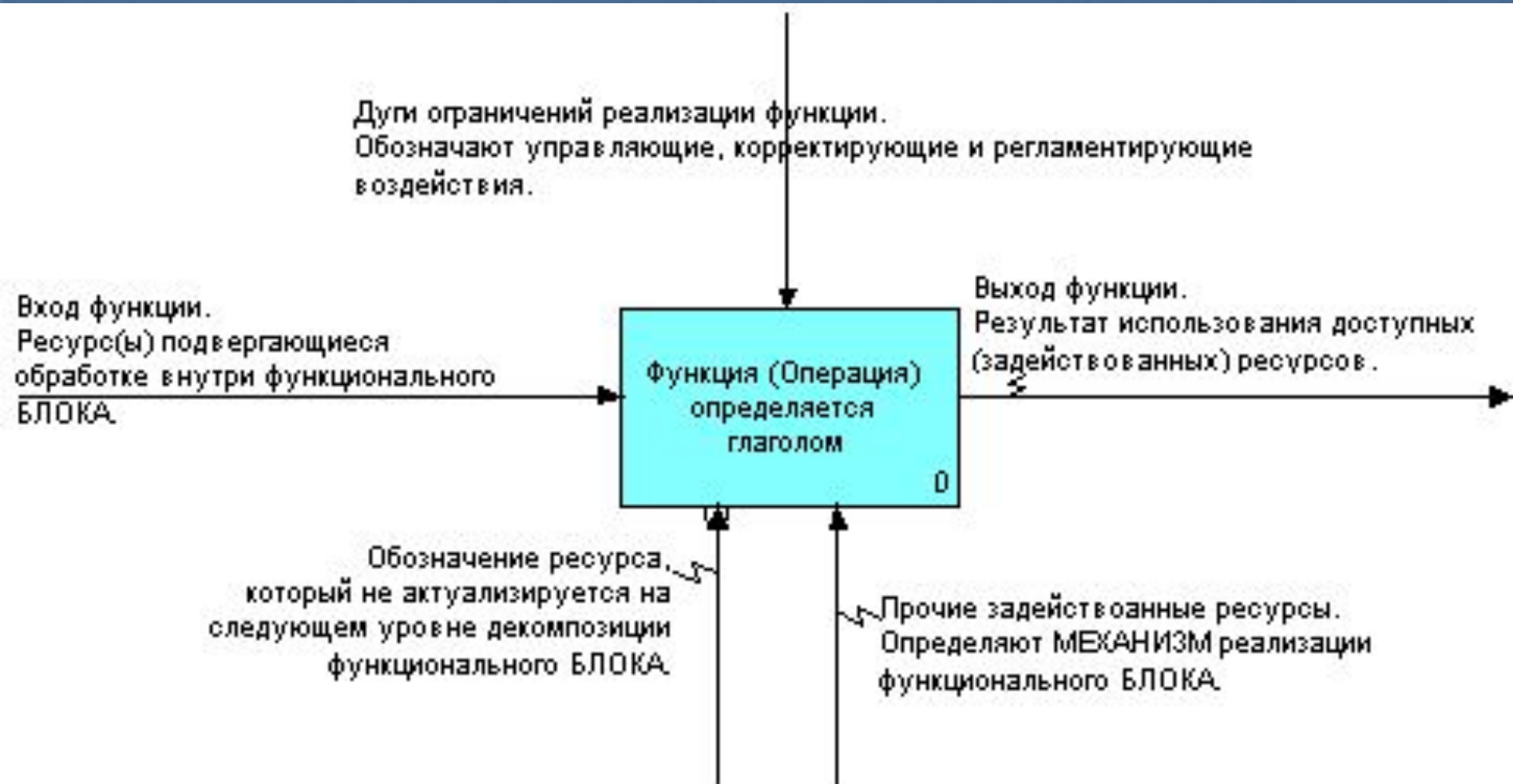
Механизм (человек или автоматизированная система), который осуществляет функцию, представляется дугой, входящей в блок снизу (см. рис.).

### ТИПЫ ДУГ

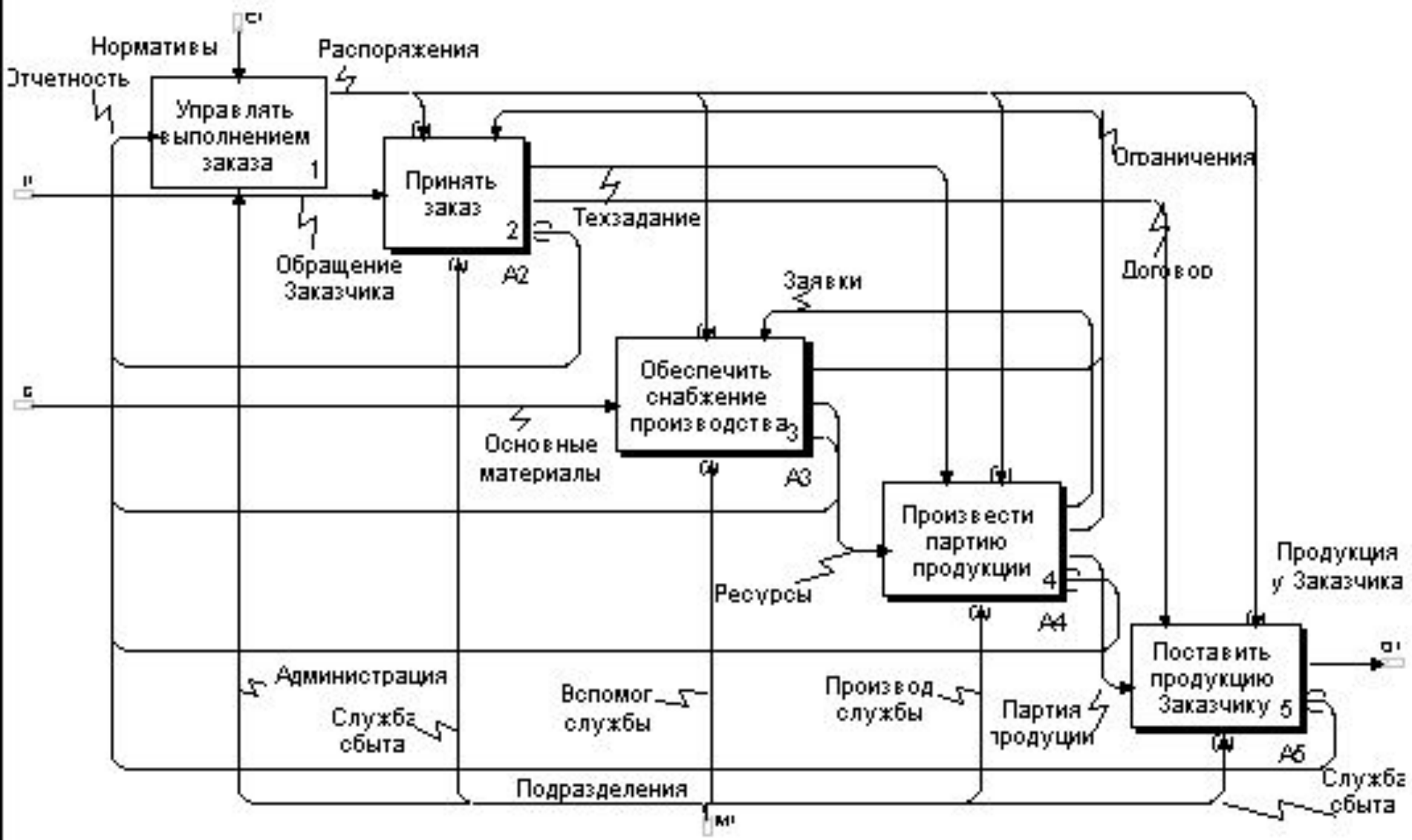


Функциональный блок преобразует входную информацию (данные, материалы, средства, задачи, цели и др.) в выходную (что требуется получить в результате выполнения данной функции). Управление определяет, когда и как это преобразование может или должно произойти. Механизм (или исполнители) непосредственно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

За каждой Дугой закрепляется Замечание, которое отображает суть информации или объекта. Замечание формулируется в виде оборота существительного, отвечающего на вопрос: "Что?".



**ДУГИ, КАК ОГРАНИЧИВАЮЩИЕ И УТОЧНЯЮЩИЕ ФАКТОРЫ БЛОКА**



**ПРИМЕР А0 ДИАГРАММЫ**



Функциональные Блоки на Диаграмме изображаются в виде прямоугольников, внутри которых записывается имя функции и номер Блок (в правом нижнем углу прямоугольника).

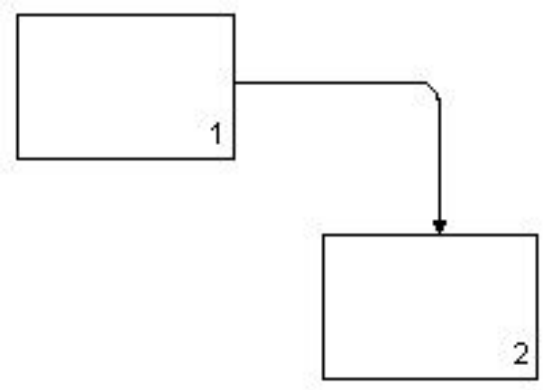
Блоки располагаются на Диаграмме согласно их степени важности (по мнению автора модели). При этом доминирующим является тот Блок, выполнение функции которого оказывает влияние на выполнение всех остальных функций, представленных на Диаграмме. К примеру, это может быть Блок, содержащий контролирующую или планирующую функцию, выходы которого являются управляющими для всех остальных функциональных Блоков Диаграммы.

Доминирующий Блок помещается, как правило, в верхнем левом углу листа Диаграммы, а наименее важный Блок – в правом нижнем углу. Таким образом, ступенчатость Блоков на Диаграмме отражает мнение автора о доминировании одних Блоков относительно других. *Очень важно помнить, что доминирование блоков на диаграмме не задаёт чёткой временной зависимости операций.*

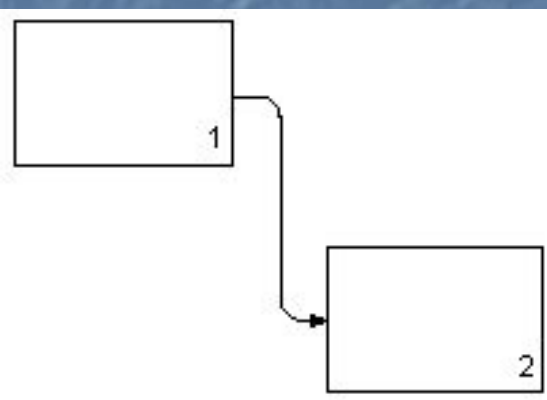
Стороны Блока также имеют определенное значение. К левой границе Блока присоединяются входные Дуги, к верхней – управляющие Дуги, к правой – выходные Дуги, а к нижней – Дуги механизмов.

Дуги на IDEF0 Диаграмме изображаются в виде стрелок.

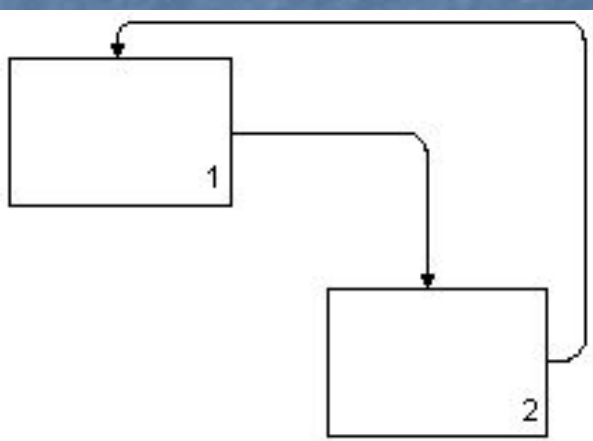
При IDEF0 моделировании используются пять типов взаимосвязей между Блоками, для описания их отношений.



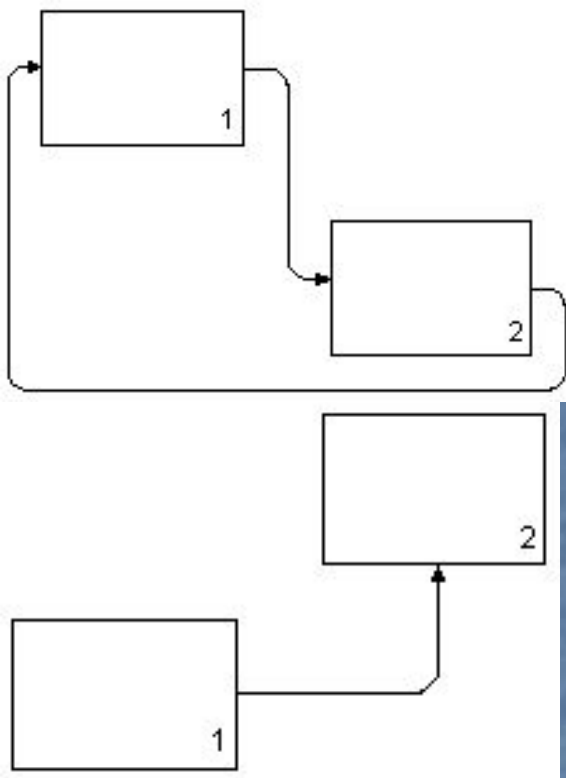
– Взаимосвязь по управлению, – когда выход одного Блока влияет (является управляющей) на выполнение функции в другом Блоке.



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



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



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

– Взаимосвязь "выход-механизм", – когда выход одной функции является механизмом для другой. Иначе говоря, выходная Дуга одного Блока является Дугой механизма для другого. Такой тип связи встречается редко и относится чаще всего к подготовительным операциям.

Поскольку содержание IDEF0 Диаграмм уточняется в ходе моделирования постепенно, Дуги на Диаграммах редко изображают один объект. Чаще всего они отображают определенный набор объектов и могут иметь множество начальных точек (источников) и определенное количество конечных точек (приемников). В ходе разработки графической Диаграммы для отражения этой особенности используют механизм разветвления/слияния Дуг. Это позволяет не только уточнить с использованием Замечаний содержание каждой ветви разветвленной Дуги (потока объектов), но и более точно описать из каких наборов объектов состоит входящая в функциональный Блок Дуга, если она получена путем слияния.

## Пример .

Рассмотрим немного надуманный пример системы – некоторый отдел контроля, созданный для оценки эффективности управления и функционирования библиотеки (с равным успехом, можно было бы также рассмотреть отдел финансового контроля на некотором предприятии, налогового контроля и т.п.) .

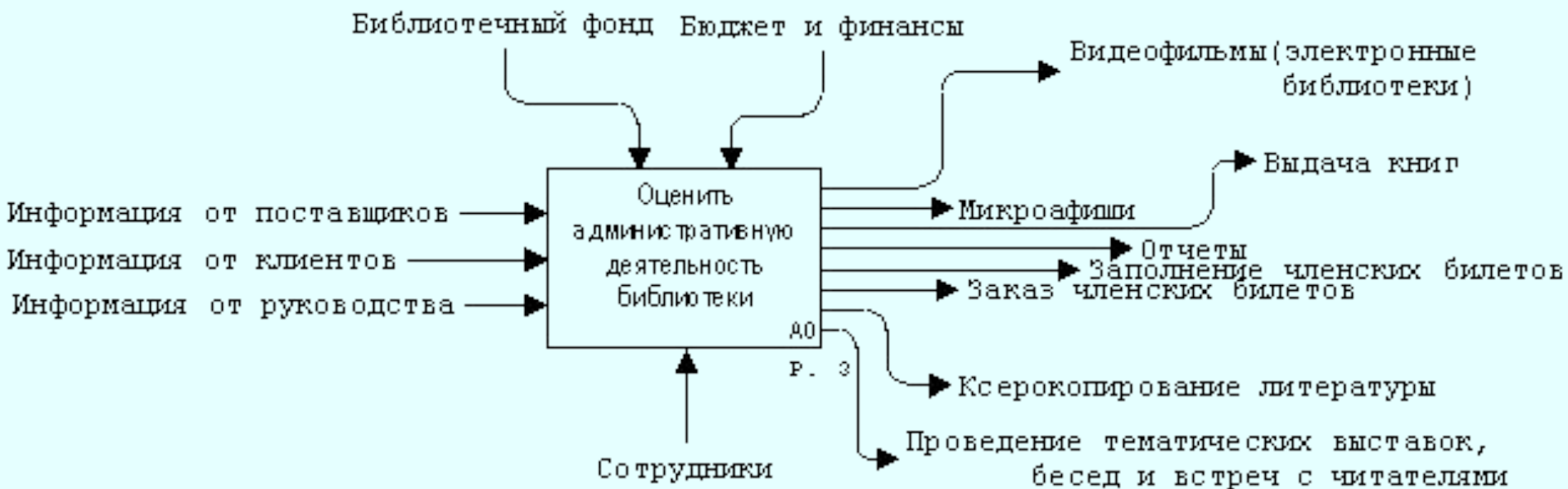


Диаграмма верхнего уровня А-0, отражающая целевую функцию системы

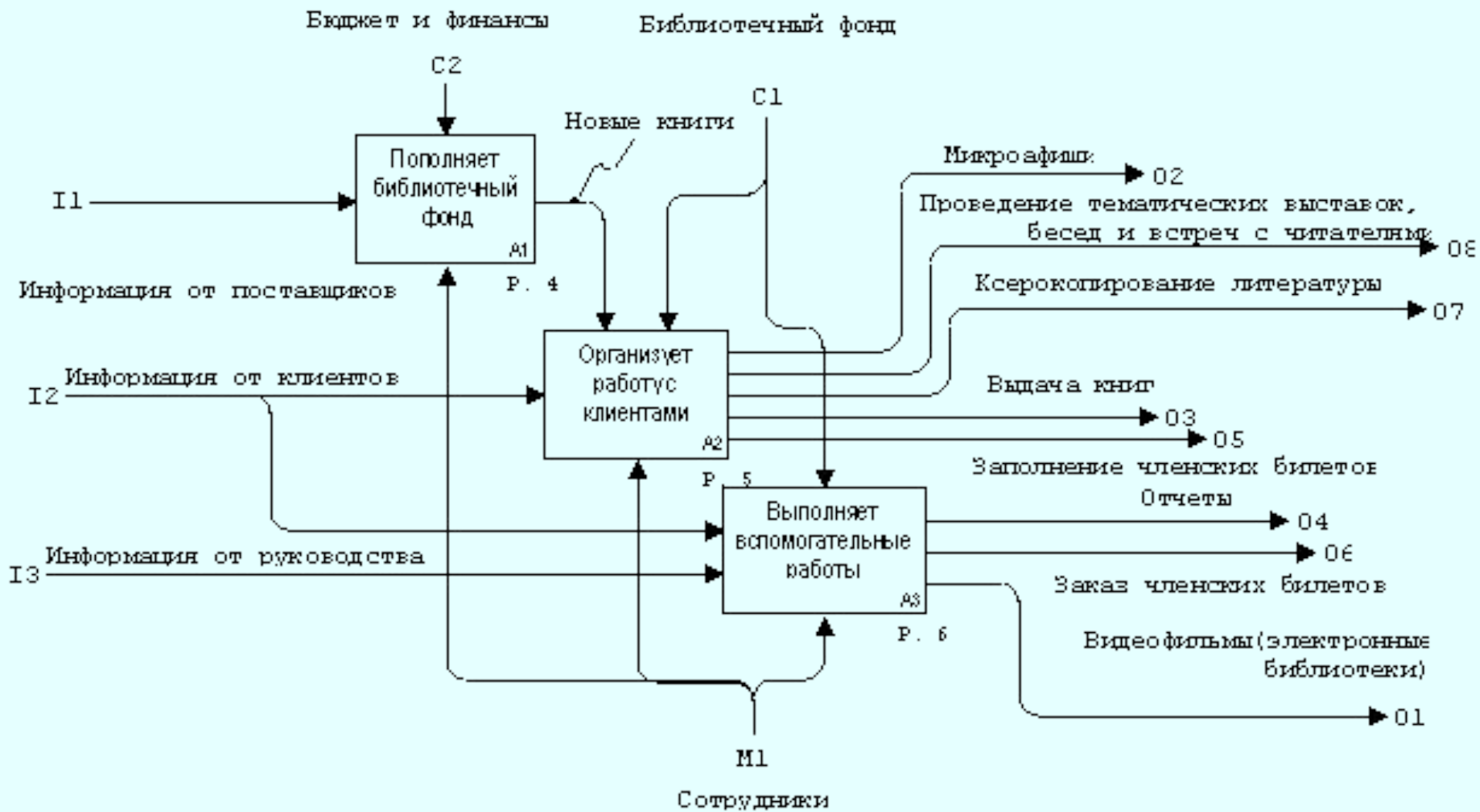
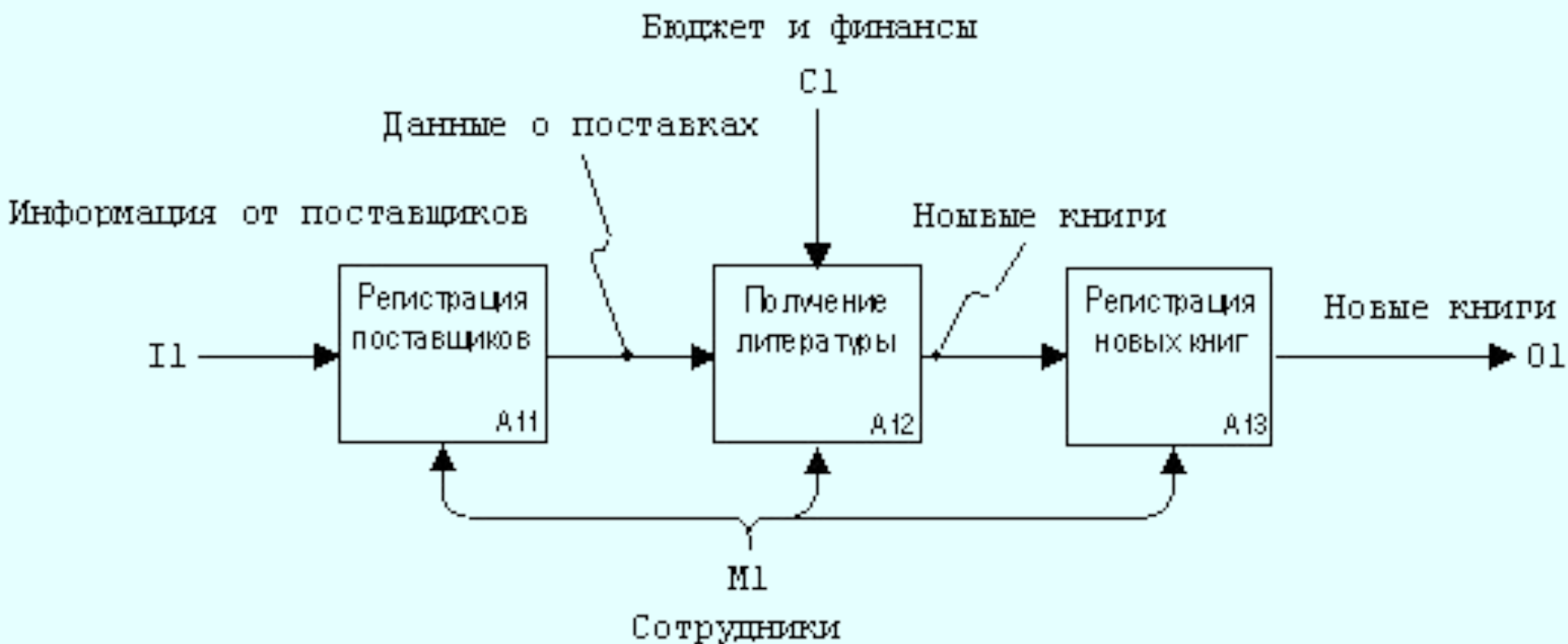
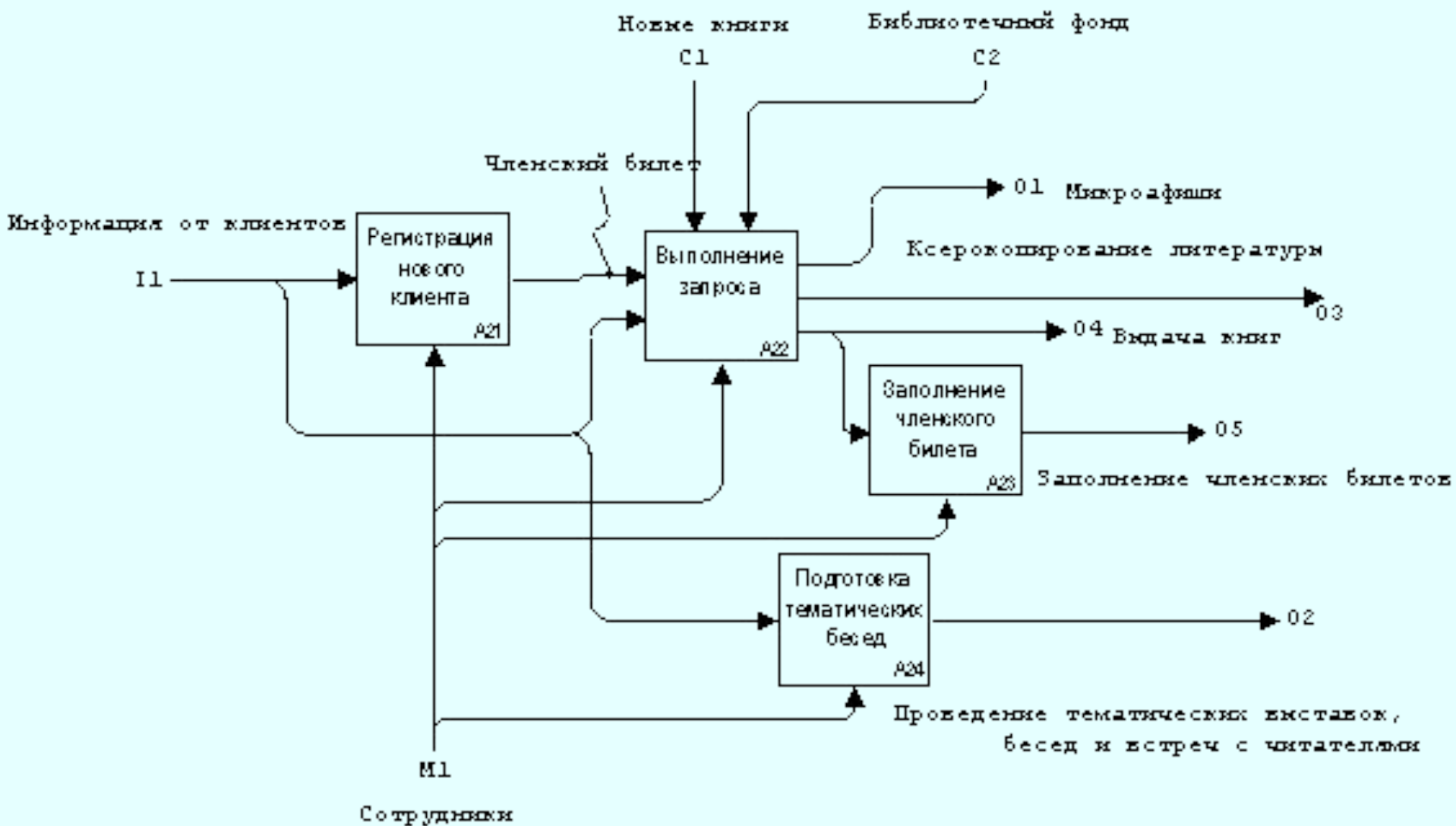


Диаграмма А0, отражающая декомпозицию целевой функции на основные функции А1, А2, А3

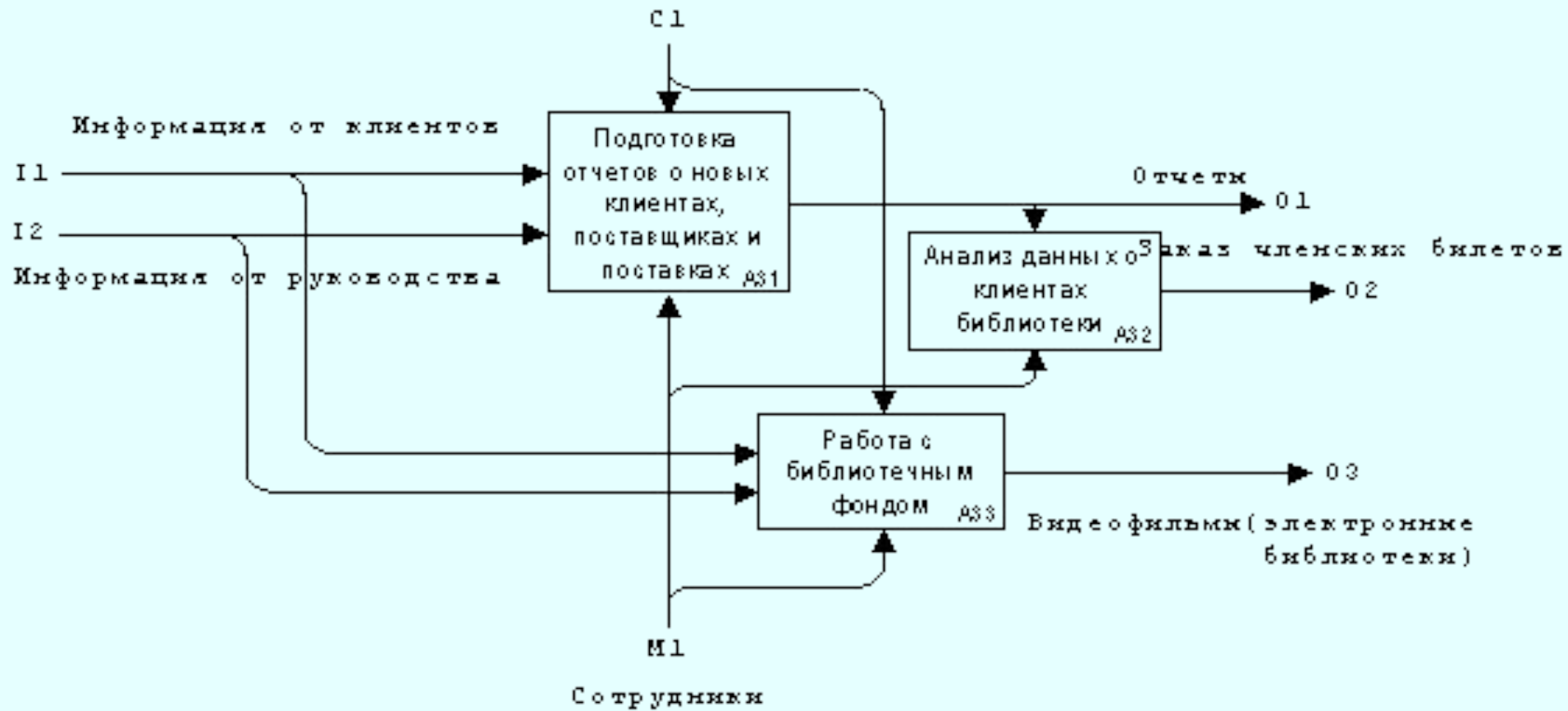


Декомпозиция блока А1



Декомпозиция блока А2

# Библиотечный фонд



Декомпозиция блока А3