МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»

Кафедра автоматизированных систем управления (АСУ)

УТВЕРЖДАЮ
Зав. кафедрой АСУ, профессор
А.М. Кориков

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ СИСТЕМ И СРЕДСТВ УПРАВЛЕНИЯ

Практические занятия и теория функционального моделирования

Учебно-методическое пособие

Уровнь основной образовательной программы: **магистратура**
Направление подготовки: **09.04.01 - Информатика и вычислительная техника**
Направленность (профиль): **Программное обеспечение вычислительных машин, систем и компьютерных сетей**

Разработчик
доцент кафедры АСУ

В.Г. Резник

2016

Резник В.Г.

Автоматизация проектирования систем и средств управления. Практические занятия и теория функционального моделирования. Учебно-методическое пособие. – Томск, ТУСУР, 2016. – 66 с.

Учебно-методическое пособие предназначено для изучения основ теории и проведения практических занятий по дисциплине «Автоматизация проектирования систем и средств управления» для студентов уровня образовательной программы магистратура направления подготовки: 09.04.01 «Информатика и вычислительная техника».

Оглавление

Введение.....5
1 Тема 1. Основные понятия автоматизации проектирования.....6
1.1 Роль систем автоматизированного проектирования при разработке систем и средств управления.....6
1.1.1 Цели и задачи САПР.....7
1.1.2 Место САПР в «жизненном цикле изделия».....7
1.2 Классификация, обозначения и основные определения САПР.....8
1.2.1 Структура САПР.....8
1.2.2 Компоненты и средства обеспечения САПР.....9
1.2.3 Классификация САПР по отраслевому назначению.....11
1.2.4 Классификация САПР по целевому назначению.....11
1.3 Предмет и задачи изучаемой дисциплины.....12
1.4 Практические занятия №1 и №2: Выбор задачи проектирования.....13
1.4.1 Общие требования к проведению практических занятий.....13
1.4.2 Критерии выбора темы проекта.....14
1.4.3 Список примеров тем проектов.....14
2 Тема 2. Основы проектирования АСУ.....15
2.1 Трехуровневая модель АСУ.....15
2.2 Понятие жизненного цикла АСУ.....17
2.2.1 Три модели ЖЦ АСУ.....17
2.2.2 Каскадная модель процесса создания АСУ.....18
2.3 Каноническое проектирование АСУ.....18
2.3.1 Группы стадий.....19
2.3.2 Группа стадий до ТЗ.....19
2.3.3 Подписание ТЗ.....20
2.3.4 Исполнение системы.....21
2.3.5 Завершение работ.....21
2.4 Практические занятия №3 и №4: Описание прикладного содержания задачи проектирования.....21
3 Тема 3. Функциональное моделирование АСУ.....22
3.1 Структурное моделирование как парадигма исследования систем управления.....22
3.1.1 Два вида моделей («as is» и «to be»).....22
3.1.2 Методологии описания предметной области: функциональный и объекто-ориентированный подходы.....23
3.1.3 Стандартизация функционального подхода для исследования систем управления (стандарт IDEF0).....24
3.1.4 Базовые понятия стандарта IDEF0.....25
3.1.5 Компьютерные инструментальные средства функционального описания и исследования систем управления.....27
3.2 Практические занятия №5 и №6: Создание контекстной диаграммы модели IDEF0.....27
3.3 Синтаксис и семантика графического языка IDEF0.....28

3.3.1 Синтаксис языка IDEF0.....28
3.3.2 Семантика языка IDEF0.....29
3.4 Практическое занятие №7: «ЦЕЛЬ» и «ТОЧКА ЗРЕНИЯ» проекта.....34
3.5 Отношения блоков на диаграммах IDEF0.....35
3.5.1 Отношения доминирования.....35
3.5.2 Отношение управления.....35
3.5.3 Отношение выход-вход.....36
3.5.4 Обратная связь по управлению.....37
3.5.5 Обратная связь по входу.....37
3.5.6 Связь «вход-механизм».....38
3.6 Уточняющие элементы диаграмм IDEF0.....39
3.6.1 ICOM-кодирование граничных стрелок.....39
3.6.2 Стрелки, помещенные в «туннель».....40
3.6.3 Перечень узлов.....41
3.6.4 Дерево узлов.....42
3.7 Практическое занятие №8: Декомпозиция модели проекта.....43
3.8 Методические приемы разработки моделей IDEF0.....44
3.8.1 Обобщенные определения.....44
3.8.2 Классификация функций.....44
3.8.3 Четыре основных вида функций.....45
3.8.4 Дополнительные виды функций.....46
3.8.5 Организационно-технические структуры и механизмы.....46
3.8.6 Управление.....47
3.9 Практическое занятие №9: Завершение декомпозиции модели.....48
3.10 Состав участников проекта и их взаимодействие.....49
3.11 Практическое занятие №10: Рабочее место ОС УПК АСУ.....50
3.12 Практическое занятие №11: Настройка среды ОС УПК АСУ.....50
3.13 Стандартный бланк методологии IDEF0.....51
3.13.1 Область проектной информации.....51
3.13.2 Область идентификации.....53
3.14 Практическое занятие №12: Изучение среды разработки RAMUS….53
3.15 Практическое занятие №13: Создание проекта в среде RAMUS.....53
3.16 Практическое занятие №14: Создание макета отчета по проекту…..54
3.17 Функциональное моделирование в методике IDEF3.....54
3.17.1 Синтаксис IDEF3.....56
3.17.2 Семантика IDEF3.....57
3.18 Методология DFD.....60
3.18.1 Синтаксис DFD.....61
3.18.2 Семантика DFD.....61
3.19 Практическое занятие №15: Предварительная защита отчета.....63
3.20 Практическое занятие №16: Защита проекта системы.....63
Заключение.....64
Список использованных источников.....65

Введение

Данное учебно-методическое пособие входит в состав учебно методического обеспечения дисциплины «Автоматизация проектирования систем и средств управления» изучаемого студентами кафедры АСУ ТУСУР уровня образовательной программы магистратура по направлению подготовки: 09.04.01 «Информатика и вычислительная техника».

Востребованность данной дисциплины обусловлена сложностью современных автоматизированных систем независимо от того, относятся ли они, в целом к системам АСУ предприятия, или только к отдельным их подсистемам. В любом случае, актуальным становится вопрос автоматизации проектирования таких технических решений, которые бы обеспечили реальную реализуемость и обоснованность создаваемых проектов. Поскольку современный тренд реализации сложных программных систем неуклонно смещается в область объектно-ориентированного программирования (ООП), возникает острая необходимость дополнять образование студентов альтернативными методами, которые используют функциональный подход.

Целью данной дисциплины является формирование у студента теоретических знаний и практических навыков, направленных на функциональное моделирование элементов систем, относящихся к классу автоматизированных систем управления (АСУ).

Задачами изучения дисциплины являются:

- овладение студентами теоретическими основами декомпозиции целевых функций систем или подсистем АСУ, а также определение необходимого уровня такой декомпозиции;
- освоение, в процессе функционального моделирования, инструментальных средств автоматизированного проектирования;
- обучение процессам и методам документирования проектных решений, в соответствии со стандартом IDEF0.

Учебный материал данного пособия представлен в виде трех разделов, охватывающих все аспекты обозначенные для достижения поставленных задач:

- Тема 1 - содержит необходимые сведения и понятия касающиеся общих аспектов автоматизации проектирования, достаточно полно изложенных в учебниках [1-2];
- Тема 2 - уточняет область проектирования, применительно к уровню образовательной программы и тематическому направлению кафедры АСУ;
- Тема 3 — содержит исчерпывающие сведения по функциональному моделированию систем, на основе модели IDEF0, а также краткое изложение дополнительных моделей: IDEF3 и DFD.

Теоретический материал данного пособия подкрепляется полным набором практических занятий, реализуемых на вычислительной инфраструктуре кафедры АСУ и обеспечивающих формирование реальных практических навыков студентов.

1 Тема 1. Основные понятия автоматизации проектирования

Современные системы автоматизированного управления (АСУ) — сложны как по составу технических и программных средств, так и по многообразию выполняемых ими функций управления.
В сложившейся ситуации, актуальным подходом является использование специализированных систем, известных как «*Системы автоматизированного проектирования* (САПР)».

Данная тема полностью посвящена общим вопросам, связнным с понятием САПР и сопутствующему ей концептуальному и техническому обеспечению.

1.1 Роль систем автоматизированного проектирования при разработке систем и средств управления

Деятельность человека как проектировщика отличается от деятельности научного исследователя тем, что:

- *научный исследователь* создает новое научное знание;
- *проектировщик*, в свой деятельности, опирается на уже полученные знания и ориентирован на создание новых систем, на базе уже известных.

Автоматизированное проектирование предполагает, в процессе разработки и принятия проектных решений, использование вычислительной техники, и программных систем, ориентированных подготовку проектной документации. Соответственно, **системы программно-аппаратных комплексов**, используемых человеком при проектировании систем, называются **САПР**, или *– системы автоматизированного проектирования*.

Замечание

Следует отметить, что сами САПР автоматизируют лишь отдельные процессы, выполняемые проектировщиком.
Центральным звеном процесса создания системы остается сам человек-проектировщик, знающий:

- объект проектирования;
- цель проектирования;
- этапы и методы проектирования;
- владеющий инструментами проектирования.

Таким образом, **целевое назначение САПР** — обеспечение коллектива проектировщиков *инструментальными средствами создания сложных систем*.

САПР, автоматизирующие работу проектировщика, должны иметь уже готовые:

- *модели* процессов, происходящих в системе;
- *модели* входных и выходных объектов и сигналов;
- *шаблоны* управления этими процессами и входными сигналами.

Замечание

Учитывая сложность и целевое многообразие АСУ, невозможно создать САПР общего проектирования:

- *все существующие САПР* являются специализированными.
- *соответственно*, специализированными являются и проектировщики, создающие АСУ.

Основные положения САПР изложены в *ГОСТ 23501.101-87*.

1.1.1 Цели и задачи САПР

Основная цель создания САПР - повышение эффективности труда инженеров, включая:

- *сокращения трудоемкости* проектирования и планирования;
- *сокращения сроков* проектирования;
- *сокращения себестоимости* проектирования и изготовления;
- *уменьшение затрат* на эксплуатацию;
- *повышения качества* и *технико-экономического уровня* результатов проектирования;
- *сокращения затрат* на натурное моделирование и испытания.

Задачами, для достижения указанных целей, являются:

- *автоматизация оформления* документации;
- *информационная поддержка и автоматизация* процессов принятия решений;
- *использование технологий* параллельного проектирования;
- *унификация проектных решений и процессов* проектирования;
- *повторное использование* проектных решений, данных и наработок;
- *стратегическое проектирование*;
- *замена натурных испытаний и макетирования* математическим моделированием;
- *повышение качества управления* проектированием;
- *применение методов* вариантного проектирования и оптимизации.

1.1.2 Место САПР в «жизненном цикле изделия»

Производственная ориентация систем АСУ требует учета теоретических представлений, связанных с концепцией «*жизненного цикла изделя*».

В рамках «жизненного цикла изделий», САПР решает задачи *автоматизации стадий проектирования* и *подготовки производства*.

Соответственно, в структуре САПР, выделяют следующие элементы:

компоненты обеспечения,
подсистемы,
ПМК – программно-методические комплексы,
ПТК – программно-технические комплексы,
КСАП - комплекс средств автоматизации проектирования.

1.2 Классификация, обозначения и основные определения САПР

САПР - *автоматизированная система*, реализующая информационную технологию выполнения функций проектирования.

САПР представляет собой *организационно-техническую систему*, предназначенную для автоматизации процесса проектирования, и состоящую из:

- *персонала*;
- *комплекса технических, программных* и других средств автоматизации деятельности этого персонала.

В английском языке термину *САПР* **соответствует** термин *CAD* (*computer-aided design*), подразумевающий использование компьютерных технологий в проектировании.

Термин САПР на английский язык может также переводится как:

- *CAD system;*
- *automated design system;*
- *CAE system*.

1.2.1 Структура САПР

Составными структурными частями *САПР* являются *подсистемы*, обладающие всеми свойствами систем и создаваемые как самостоятельные системы.

Каждая подсистема - это выделенная по некоторым признакам *часть САПР*, обеспечивающая выполнение некоторых функционально-законченных последовательностей проектных задач, с получением соответствующих проектных решений и проектных документов.

По назначению подсистемы САПР разделяют на два вида: *проектирующие* и *обслуживающие*:

- *Проектирующие подсистемы* - объектно-ориентированные подсистемы, реализующие определенный этап проектирования или группу связанных проектных задач. В зависимости от отношения к объекту проектирования они делятся на ***объектные*** и ***инвариантные***.

- ***Объектные*** **-** выполняющие проектные процедуры и операции, непосредственно связанные с конкретным типом объектов проектирования.
- ***Инвариантные*** - выполняющие унифицированные проектные процедуры и операции, имеющие смысл для многих типов объектов проектирования.

- *Обслуживающие подсистемы* - объектно-независимые подсистемы реализующие функции, общие для подсистем или САПР в целом; они обеспечивают функционирование проектирующих подсистем, оформление, передачу и вывод данных, сопровождение программного обеспечения и другие; их совокупность называют *системной средой (или оболочкой)* **САПР**.

1.2.2 Компоненты и средства обеспечения САПР

Каждая подсистема, в свою очередь состоит из *компонентов*, обеспечивающих функционирование подсистемы.
Компонент выполняет определенную функцию в подсистеме и представляет собой наименьший (неделимый) самостоятельно разрабатываемый или покупной элемент **САПР**: программа, файл модели транзистора, графический дисплей, инструкция и другие.
Совокупность однотипных компонентов образует *средство обеспечения САПР*.

Выделяют следующие *виды обеспечения САПР*:

1. ***Техническое обеспечение (ТО)*** **-** совокупность связанных и взаимодействующих *технических средств*, обеспечивающих работу **САПР**, включающая различные средства.
2. ***Математическое обеспечение (МО)***, объединяет математические методы, модели и алгоритмы, используемые для решения задач автоматизированного проектирования. МО по назначению и способам реализации делят на две части:
   - *математические методы* и построенные на их основе математические модели объектов проектирования или их части;
   - *формализованное описание* технологии автоматизированного проектирования.
3. ***Программное обеспечение (ПО)***, представляет компьютерные программы, необходимые для осуществления процесса проектирования (см. рис. 1.1). Оно подразделяется на ***общесистемное*** и ***прикладное***:
   - *общесистемное ПО*, предназначеное для управления компонентами технического обеспечения и обеспечения функционирования прикладных программ. Примером является операционная система.
   - *прикладное ПО*, реализующее математическое обеспечение для непосредственного выполнения проектных процедур. ОНО включает пакеты прикладных программ, предназначенные для обслуживания определенных этапов проектирования или групп однотипных задач внутри различных этапов.

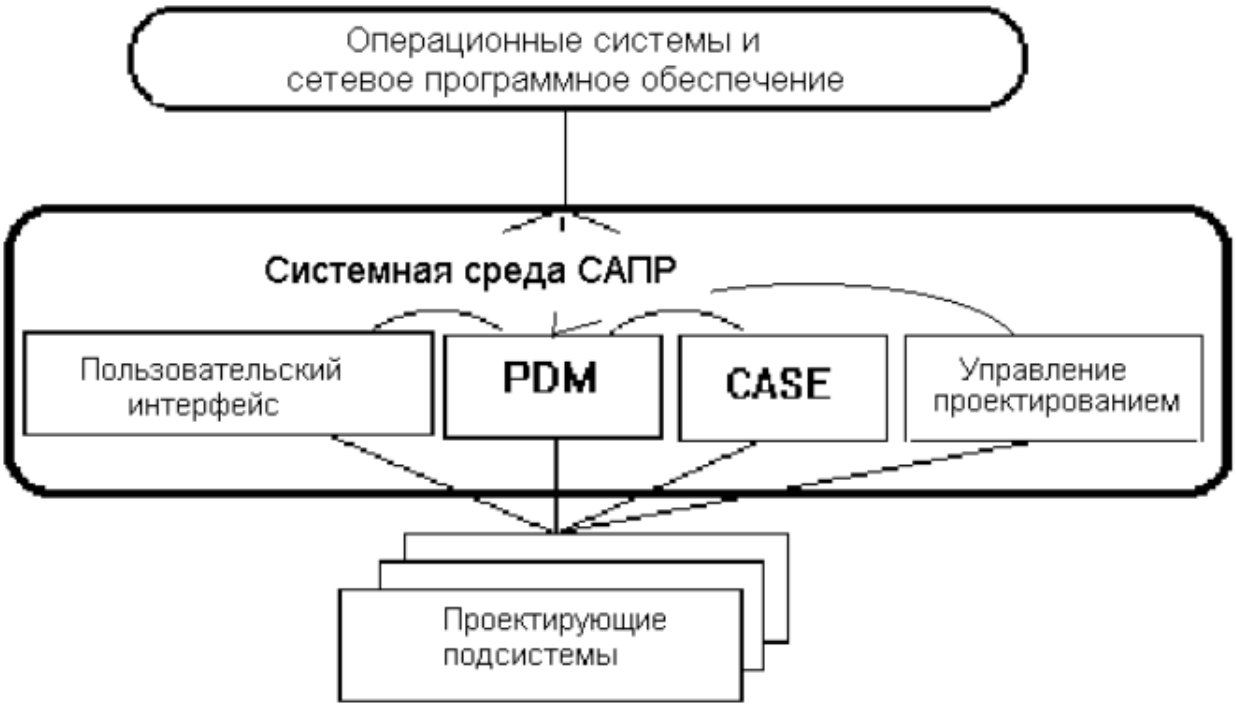

Рисунок 1.1 — Программное ПО САПР
(PDM — Product Data Management)

4. ***Информационное обеспечение (ИО)*** - совокупность сведений, необходимых для выполнения проектирования. Состоит из *описания стандартных проектных процедур, типовых проектных решений, комплектующих изделий и их моделей, правил и норм проектирования*. Основная часть ***ИО*** **САПР —** базы данных и СУБД.
5. ***Лингвистическое обеспечение (ЛО)*** - совокупность языков, используе-мых в САПР для представления информации о проектируемых объектах, процессе и средствах проектирования, а также для осуществления диалога проектировщик-ЭВМ и обмена данными между техническими средствами САПР. Включает *термины, определения, правила формализации естест-венного языка, методы сжатия и развертывания*. В ***ЛО*** выделяют классы различных типов языков проектирования и моделирования: *VHDL, VERILOG, UML, GPSS*.
6. ***Методическое обеспечение (МетО)*** - описание технологии функционирования САПР, методов выбора и применения пользователями технологических приемов для получения конкретных результатов. Включает в себя *теорию процессов, происходящих в проектируемых объектах, методы анализа, синтеза систем и их составных частей, различные методики проектирования*. Иногда к **МетО** относят также ***МО*** и ***ЛО***.
7. ***Организационное обеспечение (ОО)*** - совокупность документов, определяющих состав проектной организации, связь между подразделениями, организационную структуру объекта и системы автоматизации, деятельность в условиях функционирования системы, форму представления результатов проектирования. В ***ОО*** входят *штатные расписания, должностные инструкции, правила эксплуатации, приказы, положения и другое*.

Замечание

В САПР выделяют также *эргономическое* и *правовое* обеспечения:

- *Эргономическое обеспечение* объединяет взаимосвязанные требования, направленные на согласование психологических, психофизиологических, антропометрических характеристик и возможностей человека с техническими характеристиками средств автоматизации и параметрами рабочей среды на рабочем месте.
- *Правовое обеспечение* состоит из правовых норм, регламентирующих правоотношения при функционировании **САПР**, и юридический статус результатов ее функционирования.

1.2.3 Классификация САПР по отраслевому назначению

В зависимости от отраслевого назначения выделяют:

- ***MCAD*** (*mechanical computer-aided design*) - автоматизированное проектирование механических устройств. Машиностроительные **САПР**, применяются в автомобилестроении, судостроении, авиакосмической промышленности, производстве товаров народного потребления. Они включают в себя разработку деталей и сборок (механизмов) с использованием параметрического проектирования на основе конструктивных элементов, технологий поверхностного и объемного моделирования: *SolidWorks, Autodesk Inventor, CATIA*;
- ***EDA*** (*electronic design automation*) или ***ECAD*** (*electronic computer-aided design*) - **САПР** электронных устройств, радиоэлектронных средств, печатных плат и другие: Altium Desinger, OrCAD;
- ***AEC CAD*** (*architecture, engineering and construction computer-aided design*) или ***CAAD*** (*computer-aided architectural design*) - **САПР** в области архитектуры и строительства. Используются для проектирования зданий, промышленных объектов, дорог, мостов и другие: Autodesk Atchitectural Desktop, Piranesi, ArchCAD.

1.2.4 Классификация САПР по целевому назначению

По целевому назначению **САПР**, или подсистемы **САПР**, различают по аспектам проектирования:

- ***CAD*** (*computer-aided design/drafting*) - средства автоматизированного проектирования. В контексте указанной классификации термин обозначает средства **САПР,** предназначенные для автоматизации двумерного и/или трехмерного геометрического проектирования, создания конструкторской и/или технологической документации, **САПР** общего назначения. Для обозначения данного класса средств **САПР** используется также термин ***CADD*** (*computer-aided design and drafting*) - автоматизированное проектирование и создание чертежей. Системы геометрического моделирования обозначают как ***CAGD*** (*computer-aided geometric design*).
- ***CAE*** (*computer-aided engineering*) - средства автоматизации инженерных расчётов, анализа и симуляции физических процессов. Они осуществляют

динамическое моделирование, проверку и оптимизацию изделий. Подкласс средств CAE, используемых для компьютерного анализа, обозначается термином ***CAA*** (*computer-aided analysis*).

- ***CAM*** (*computer-aided manufacturing*) - средства технологической подготовки производства изделий. Они обеспечивают автоматизацию программирования и управления оборудованием с **ЧПУ** или **ГАПС** (Гибких автоматизированных производственных систем). Русским аналогом термина является **АСТПП** - автоматизированная система технологической подготовки производства.
- ***CAPP*** (*computer-aided process planning*) - средства автоматизации планирования технологических процессов, применяемые на стыке систем **CAD** и **CAM**.

Замечание
Многие системы автоматизированного проектирования совмещают в себе решение задач относящихся к различным аспектам проектирования *CAD/CAM*, *CAD/CAE* или *CAD/CAE/CAM*.
Такие системы называют *комплексными или интегрированными*.

1.3 Предмет и задачи изучаемой дисциплины

Учитывая многобразие систем автоматизированного проектирования, выраженные в различиях как по отраслевому, так и по целевому назначениям, уточним цели и задачи, решаемые в пределах данной дисциплины.

Общей целью преподавания дисциплины является обучение студентов основным *принципам, способам и методам автоматизации проектирования*, необходимым при создании систем и средств управления.
Указанные знания достаточно полно изложены в учебниках [1-2].

Непосредственной целью обучения студентов является получение знаний, навыков и умений *функционального моделирования автоматизированных систем* (АС) широкого назначения, что опирается на освоение трех методологий: IDEF0, IDEF3 и DFD.
Основной учебный материал хорошо описан в учебнике [2].
Дополнительный учебный материал содержится в ГОСТ 34.601-90, ГОСТ 34.602-89, ГОСТ 19.102-77, ГОСТ 19.201-78, ГОСТ Р 50.1.028-2001.

Задачи изучения дисциплины:

- освоить методику IDEF0 применительно к *функциональному моделированию* автоматизированных систем управления *(АСУ)*;
- освоить систему разработки RAMUS или ErWin;
- закрепить полученные знания созданием конкретного проекта функциональной модели организационной системы.

Учебный материал для указанных задач содержится в данном учебном пособии.

1.4 Практические занятия №1 и №2: Выбор задачи проектирования

Методика обучения по дисциплине «Автоматизация проектирования систем и средств управления» предполагает выполнение практических занятий в объеме ***32 учебных часа***.

Учебные занятия проводятся в учебных классах кафедры АСУ обеспеченных:

- вычислительной техникой для индивидуальной работы каждого студента;
- средствами автоматизации проектированиядля ведения каждым студентом индивидуальной разработки (проекта) системы.

С целью более эффективного процесса обучения и усвоения учебного материала, каждый студент:

- выполняет отдельную проектную работу;
- оформляет личный индивидуальный отчет;
- проводит коллективное обуждение достигнутых результатов;
- проводит индивидуальную защиту своего отчета.

Таким образом, достигается должный уровень подготовки специалиста.

Первое практическое занятие посвящено выбору индивидуальной темы проектирования, которое проводится в три этапа:

- усваиваются *общие требования* проведения практических занятий;
- осознаются *критерии выбора* темы проектирования;
- непосредственный *выбор темы* проектирования, проходящий процедуру индивидуального контроля.

1.4.1 Общие требования к проведению практических занятий

Общая концепция режима проведения практических занятий - *групповое обучение* с индивидуальным контролем выполнения каждого проекта.

Общие требования:

- *выбор студентом* индивидуальной темы проектирования, которая разрабатывается на всех практических занятиях;
- *последовательная разработка темы* на каждом практическом занятии в соответствии с этапами проектирования;
- *индивидуальное согласование с преподавателем* каждого этапа проектирования в соответствии с темой практического занятия;
- *оформление результатов проектирования* в виде индивидуального отчета;
- *индивидуальная защита* результата проектирования.

1.4.2 Критерии выбора темы проекта

Выбираемая тема проектирования должна предполагать, что студент будет проводить *функциональное моделирование* системы, являющейся *частью автоматизированной системы управления* (АСУ) некоторой организации (предприятия).

Выбираемая система проектирования должна быть:

- способной формулироваться как *обобщенная функция*, допускающая минимум три уровня декомпозиции;
- *иметь входы*, в виде информационных и/или материальных объектов;
- *иметь выходы*, в виде информационных и/или материальных объектов;
- *иметь механизмы* — люди, реализующие функцию;
- *иметь управление* — инструкции, приказы, распоряжения и другое, по ограничениям которых функционирует система.

Замечание

Желательно, чтобы студент сам имел опыт функционирования в системе в качестве механизма.

1.4.3 Список примеров тем проектов

Тему проекта следует формулировать, указывая *два уровня названий*:

- *имя* проектируемой системы;
- *механизм* (сотрудник, сотрудники), реализующий функции системы.

Например:

1. Отдел тестирования программного обеспечения. Программист, обеспечивающий тестирование ПО.
2. Отдел продаж магазина. Продавец-кассир, реализующий товар.
3. Бюро проектирования сайтов. Разработчик ПО сайтов.
4. B2B-центр. Исполнитель отчета для «Консультант+».

Замечание

Выбрав тему, студент проводит ее краткое словесное описание в личном отчете.

Практическое занятие №2 проходит по тематике первого занятия.
Результат занятия — список утвержденных индивидульных тем проектов.

2 Тема 2. Основы проектирования АСУ

Данная тема посвящена основам теоретических концепций объекта проектирования:

- трехуровневой модели АСУ;
- понятию жизненного цикла АСУ;
- каноническому подходу к проектированию АСУ.

2.1 Трехуровневая модель АСУ

Рассматривая любую организацию, фирму, институт или любое юридическое лицо, можно говорить об *АСУ такого объекта*.

Традиционно выделяется *три иерархически связанных уровня АСУ*.

Уровень управления	***Уровень АСУ***
Уровень стратегического управления	**АСУП** – *АСУ производством* **или ERP**-системы. Стратегическое административно-финансовое планирование и управление, решающее задачи: *что произвести*, в каких объемах, к каким срокам, из чего и прочее. **ERP** - *Enterprise Resource Planning*, планирование ресурсов предприятия.
Уровень тактического управления	**АСУПП** - *АСУ производственными процессами* **или MES**-системы**.** Уровень начальников производств, цеховых технологов, диспетчеров, мастеров, решающих задачу: *как произвести заданное*, по каким технологиям, на каких станках, в каком порядке выполнять заказы, чтобы минимизировать издержки и максимально эффективно использовать ресурсы. **MES** - *manufacturing execution system*, система управления производственными процессами.
Уровень оперативного исполнения	**АСУТП** – *АСУ технологическими процессами.* Уровень контроллерного управления, HMI с человеком исполнителем, **SCADA**-системы - решает задачи: *поддержания заданных технологических параметров производственных процессов*. **SCADA** - *supervisory control and data acquisition*, диспетчерское управление и сбор данных.

Верхний и нижний уровни иерархии управления АСУ достаточно хорошо прорабо-аны как в теоретическом плане, так и в плане практической реализации:

- **АСУПП** достаточно полно регламентируется и управляется набором государственных постановлений и указов о делятельности предприятий и организаций; для реализации этого уровня управления используются системы **ERP**;
- **АСУТП** также имеет свою теоретическую базу, основанную на различных технических решениях, и реализуется в виде систем **SCADA**.

Средний уровнь иерархии управления АСУ, до настоящего времени, не имеет некоторой эталонной системы, даже несмотря на то, что создана международная огранизация **MESA**. Рассмотрим этот уровень подробнее.

Ассоциация MESA International была *основана в 1992 году* с названием *Manufacturing Execution System Association* - как ассоциация поставщиков и пользователей MES-систем.
Основной целью, на момент создания, было информирование производителей о системах отслеживания выполнения заказов на производстве.

В 2001 году ассоциация сменила свое название на *Manufacturing Enterprise Solutions Association*, чтобы показать, что область интересов ассоциации включает всё программное обеспечение, используемое на производстве.
Целью ассоциации стал обмен передовым опытом и инновационными идеями для распространения знаний о решениях в области оперативного управления производственными предприятиями.

MESA является автором «*Функциональной модели MES*», включающей основные функции систем класса MES.
Первоначально, по модели 1997 года, она **включала 11 функций**.
В 2004 году их число было **сокращено до 8**:

- Контроль состояния и распределение ресурсов (RAS)
- *Оперативное/Детальное планирование (ODS)* - *удалено в модели 2004 года*;
- Диспетчеризация производства (DPU) ;
- *Управление документами (DOC)* - *удалено в модели 2004 года*;
- Сбор и хранение данных (DCA);
- Управление персоналом (LM);
- Управление качеством продукции (QM);
- Управление производственными процессами (PM);
- *Управление производственными фондами (техобслуживание) (MM)* — *удалено в модели 2004 года*;
- Отслеживание истории продукта (PTG);
- Анализ производительности (PA).

Таким образом, несмотря на достаточную теоретическую проработанность концепции АСУ, имеется множество задач и подсистем, требующих создания проектов конкреных программных и технических реализаций объектов автоматизации.

2.2 Понятие жизненного цикла АСУ

Любая АСУ, как и любая техническая система, имеет свой жизненный цикл (ЖЦ):
АСУ задумывается;
Проектируется;
Создается;
Передается в промышленную эксплуатацию;
Эксплуатируется;
Морально устаревает;
Заменяется новой АСУ.

Модель жизненного цикла АСУ отражает состояние системы с момента осознания необходимости создания АСУ до полной ее утилизации.

Модель ЖЦ АСУ включает:
Стадии;
Результаты выполнения работ на каждой стадии;
Ключевые события или точки завершения работ и принятия решений.

2.2.1 Три модели ЖЦ АСУ

Известны три основных модели ЖЦ АСУ:

Каскадная модель – описывает классический подход к разработке систем в любых предметных областях. Она предусматривает последовательную организацию работ, которые разбиты на этапы. *Следующий этап начинается только после завершения предыдущего этапа*. Каждый этап завершается выпуском полного комплекта документации.

Итерационная модель – не предполагает четкого разбиения на этапы: система проектируется сразу на нескольких уровнях и состоит из серии коротких циклов (шагов) по планированию, реализации, изучению и действию. Здесь имеется *множество проектных решений «снизу-вверх»*, когда проектные решения по отдельным задачам объединяются в общие системные решения.

Спиралевидная модель – предполагает наличие этапов: *разработка требований*, *проектирование*, *реализация*, *тестирование*, *ввод в действие*. Но, в отличие от каскадной модели, предполагает итерационный процесс разработки АСУ. Каждая итерация представляет собой законченный цикл разработки, приводящий к выпуску внутренней или внешней версии изделия. На каждой итерации продукция совершенствуется, чтобы стать законченным изделием.

2.2.2 Каскадная модель процесса создания АСУ

Каскадная модель создания АСУ разделяется на пять последовательных эта-пов:

Разработка требований;
Проектирование;
Реализация;
Тестирование;
Ввод в действие.

Преимущество данного подхода – *четкое проектирование «сверху-вниз»*, но ошибки предыдущих этапов очень трудно исправлять.

2.3 Каноническое проектирование АСУ

Каноническое проектирование АСУ ориентировано на каскадную модель жизненного цикла системы.
Стадии и этапы такого проектирования описаны в *ГОСТ 34.601-90*:

Стадии	***Этапы работ***
1. Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС. 1.2. Формирование требований пользователя к АС. 1.3. Оформление отчёта о выполненной работе и заявки на разработку АС (тактико-технического задания).
2. Разработка концепции АС.	2.1. Изучение объекта. 2.2. Проведение необходимых научно-исследовательских работ. 2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя. 2.4. Оформление отчёта о выполненной работе.
3. Техническое задание.	Разработка и утверждение технического задания на создание АС.
4. Эскизный проект.	4.1. Разработка предварительных проектных решений по системе и её частям. 4.2. Разработка документации на АС и её части.
5. Технический проект.	5.1. Разработка проектных решений по системе и её частям. 5.2. Разработка документации на АС и её части. 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или)

	технических требований (технических заданий) на их разработку. 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации.
6. Рабочая документация.	6.1. Разработка рабочей документации на систему и её части. 6.2. Разработка или адаптация программ.
7. Ввод в действие.	7.1. Подготовка объекта автоматизации к вводу АС в действие. 7.2. Подготовка персонала. 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями). 7.4. Строительно-монтажные работы. 7.5. Пусконаладочные работы. 7.6. Проведение предварительных испытаний. 7.7. Проведение опытной эксплуатации. 7.8. Проведение приёмочных испытаний.
8. Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами. 8.2. Послегарантийное обслуживание.

2.3.1 Группы стадий

С точки зрения разработчика-проектировщика перечисленные стадии группируются по степени важности и особенностям исполнения.

Группа стадий	***Список стадий создания АСУ***
До ТЗ	1-Формирование требований к АС 2-Разработка концепции АС
Подписание ТЗ	3-Техническое задание
Исполнение системы	4-Эскизный проект 5-Технический проект 6-Рабочая документация (Рабочий проект)
Завершение работ	7-Ввод в действие 8-Сопровождение АС

2.3.2 Группа стадий до ТЗ

Первые две стадии «**До ТЗ**», как правило, ведутся не на договорной основе, а на инициативе «*Потенциального заказчика*» и «*Потенциальных исполнителей*».

Потенциальный заказчик формирует требования к АС и ее концепцию с помощью своих специалистов и сторонних консультантов, а также определяет список потенциальных исполнителей.

Потенциальные исполнители знакомятся с результатами *Потенциального заказчика*, оценивают свои возможности, предлагают свое видение проблем и пытаются оценить стоимость возможного проекта и свое участие в нем.
В большинстве случаев, *потенциальные исполнители* работают с некоторыми эскизами ТЗ *потенциального заказчика*.

Обычно, по результатам двух первых стадий, *потенциальные исполнители* пишут **Технико-коммерческие предложения (ТКП)**, на основе которых определяются:

- *головной исполнитель;*
- *список частных исполнителей* (субподрядчиков);
- *стоимость* разработки АСУ.

2.3.3 Подписание ТЗ

На основании результатов конкурса ТКП, головной исполнитель пишет ТЗ в соответствии с ГОСТ 34.602-89 «*Техническое задание на создание автоматизированной системы*», в котором расписаны все требования к АС, определены *Разработчик*, *Субподрядчики*, *сроки и стоимость* всех последующих стадий и этапов создания системы.

Текст ТЗ излагается в стиле требований «*как должно быть*».
Все последующие документы, несмотря на то, что работа по созданию АС еще не завершена, излагаются в совершенной форме: ***выполнено***, ***разработано*** и другие.
По необходимости, составляются частные ТЗ (ЧТЗ) на АС и ЧТЗ на разработку программного обеспечения в соответствии с *ГОСТ серии 19*:

- *ГОСТ 19.102-77* «Стадии разработки»;
- *ГОСТ 19.201-78* «Техническое задание. Требования к содержанию и оформлению».

После серии согласований, все ТЗ и ЧТЗ подписываются вместе с договором на создание АС.

На стадии подписания ТЗ разрешается, при согласовании с Заказчиком:

- *Исключить* стадию «Эскизный проект»;
- *Объединить* стадии «Технический проект» и «Рабочая документация» в одну стадию «Технорабочий проект».

После подписания ТЗ, *оно становится основным документом*, в соответствии с которым проводятся все работы, сдача этапов работ, предъявление претензий и оплата работ.

2.3.4 Исполнение системы

Стадии исполнения системы – это основная работа, которую должен выполнить *Исполнитель*.
При необходимости, **Исполнитель** без согласования с **Заказчиком**, на основании внутренних распоряжений *может*:

- *Делить* стадии и этапы работ, в соответствии с ограничениями основного ТЗ;
- *Формировать* ЧТЗ на подсистемы АС и программное обеспечение.

Группа стадий *«исполнение системы»* завершается передачей заказчику разработанных **Исполнителем**:

- *Технических и программных средств АС*;
- *Рабочей документации на АС*.

2.3.5 Завершение работ

На стадии завершения работ степень и ответственность **Исполнителя** определяется **ТЗ** и условиями договора.
Исполнитель может выступать в качестве *Наблюдателя* или *Консультанта*.

2.4 Практические занятия №3 и №4: Описание прикладного содержания задачи проектирования

Тематическое назначение практического занятия **№3**: «*Анализ и описание прикладного содержания задач на проектирование*».

Изучив теоретическую часть раздела 2 (подразделы 2.1-2.3), студент переходит к анализу прикладного описания задачи.
Такое описание должно включать:

- *описание функции* прикладной системы и ее ограничения;
- *описания входных* и *выходных объектов*, с учетом функции системы;
- *описание ограничений* на реализацию основной функции системы и указание того, чем эти ограничения регламентируются (документы, приказы, рапоряжения).

Тематическое назначение практического занятия **№4**: «*Согласование прикладного содержания задач на проектирование*».
Студент, в индивидульном порядке, согласует с преподавателем прикладную часть разрабатываемого проекта системы.

3 Тема 3. Функциональное моделирование АСУ

Данная тема содержит основной теоретический материал и методические указания, обеспечивающие обучение студентов по изучаемой дисциплине.

3.1 Структурное моделирование как парадигма исследования систем управления

Научное исследование как природных, так и социальных объектов и явлений предполагает построение некоторых моделей, которые отображали бы основные свойства этих объектов и явлений, и, дальнейшем, позволяли бы делать научные и практические выводы, а, при необходимости, управлять этими объектами и явлениями.
Современные методы построения моделей ориентированы на исследования сложных систем, каковыми являются производственные предприятия, предприятия соцкультбыта, различные военные, государственные и общественные структуры.
Актуальным является проектирование и создание систем управления такими структурами, которые получили обобщенное название: *Автоматизированные системы управления* (АСУ).

Важное место при исследовании сложных систем управления является наглядное представление моделей, которыми эти системы представлены.
Словесное описание моделей в большинстве случаев не является наглядным, хотя хорошо подходит для документирования таких описаний.
Указанный недостаток словесных (текстовых) моделей описания сложных систем стимулировал создание различных *графических способов (языков)* описания моделей.
Развитие как технических, так и программных средств вычислительной техники позволило создать различное программное обеспечение, которое автоматизирует графическое отображение моделей, обеспечивая их широкое практическое применение.

3.1.1 Два вида моделей («as is» и «to be»)

В общем случае выделяется два вида моделей:

- *Модели «as is» - «как есть»*. Такие модели отражают *текущее состояние системы управления*. Использование таких моделей позволяет максимально быстро и точно познакомиться с системой управления, выявить узкие места в ней и сформулировать предложения по ее улучшению.
- *Модели «to be» - «как должно быть»*. Такие модели отражают представления о том, *какой должна быть система управления*, как она должна работать, наметить план по созданию такой системы.

Тематика нашего обучения ограничена моделями «***as is***».

3.1.2 Методологии описания предметной области: функциональный и объекто-ориентированный подходы

В 60-е годы 20-го века, одними из первых, стали развиваться *методы информационного описания предметной области*.
Развитие этих методов стимулировалось проблемами, которые порождались концептуальной парадигмой «**Программа-массив**».
Разработчики систем управления ***«устали» постоянно согласовывать*** входные и выходные данные, которые обрабатывались программами.

Возникла парадигма *проектирования информационных систем* посредством предварительного анализа предметной области, совокупного описания и согласования форматов данных, которыми будет манипулировать система управления, создания специальных программных инструментальных средств, которые будут входить в систему управления и обеспечивать эффективный доступ к данным.
Развитие этой парадигмы породило программную продукцию, названную как СУБД (системы управления базами данных).
Все современные профессинальные СУБД, такие как ***Oracle***, ***MS SQL Server***, содержат инструментальные средства для моделирования предметной области.

Другая широко известная методология описания предметной области тесно развивалась совместно с развитием ***языков программирования***.
Эта методология известна как *объектно-ориентированный подход*.
Он предлагает выделение классов обектов и методов действий, в которых участвуют объекты.
Для объектно-ориентированного подхода создан специальный графический язык универсального моделирования ***UML***.

Третья методология описания предметной области названа *функциональным подходом*.
Она позволяет отбразить систему управления *схемой технологических процессов* (функций, операций), на входах и выходах которых отображаются объекты различной природы:

- материальные и информационные объекты;
- используемые ресурсы;
- организационные единицы и другие.

Основное преимущество функционального подхода – наглядное для большинства специалистов представление о том, *что же система делает* и *для чего она предназначена*.
В большинстве случаев использование функционального подхода *должно предшествовать* применению других подходов.
Наиболее широкое применение функциональный подход получил в *реинжиниринге*, при описании бизнес-процессов предприятий.

3.1.3 Стандартизация функционального подхода для исследования систем управления (стандарт IDEF0)

Необходимость стандартизации функционального подхода обусловлена двумя основными факторами:

- *постоянное усложнение* производственно-технических и организационно-экономических систем;
- *необходимостью создания* инструментальных компьютерных средств построения функционональных моделей.

В США в конце 70-х годов была предложена и реализована *«Программа интегрированной компьютеризации производства (ICAM - Integrated Computer Aided Manufacturing)»*.
Эта программа была предложена *департаментом военно-воздушных сил США*.
Цель программы - увеличение эффективности промышленных предприятий посредством широкого внедрения компьютерных (информационных) технологий.

В рамках программы ICAM был разработан *набор стандартов IDEF (ICAM Definition)*, позволяющее исследовать структуру, параметры и характеристики производственно-технических и организационно-экономических систем.
Было создано большое количество частных методологий моделирования IDEF, многие из которых так и не нашли широкого практического применения.

Наиболее широко используются две методологии, основанные на графическом представлении систем:

- **IDEF0** - *для создания функциональной модели*, отображающей структуру и функции системы, а также потоки информации и материальных объектов, преобразуемые этими функциями;
- **IDEF1** - *для построения информационной модели*, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы.

Сама методология IDEF0 основана на подходе, получившем название *«Метод структурного анализа и проектирования (SADT - Structured Analysis & Design Technique)»*. Как стандарт - был выпущен ***в 1981 г.***

Основу этого подхода составляет графический язык описания (моделирования) систем.

В России стандартизация IDEF0 представлена документами *РД IDEF0-2000* и *Р 50.1.028-2001*.

3.1.4 Базовые понятия стандарта IDEF0

Стандарт IDEF0 основан на *четырех базовых понятиях*, два из которых показаны на рисунке 3.1.

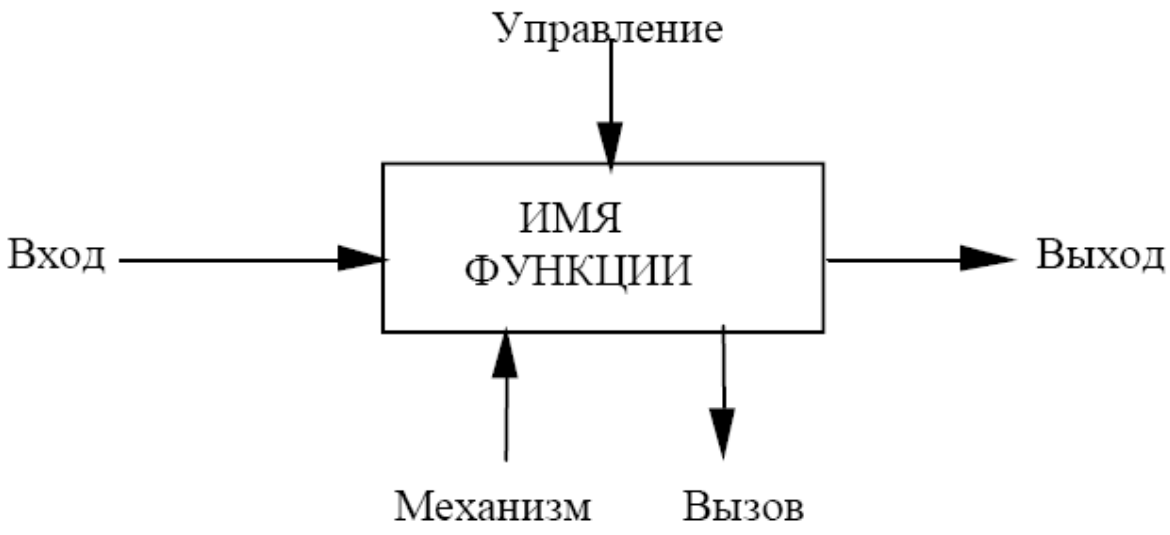

Рисунок 3.1 — Макромодель стандарта IDEF0

Функциональный блок (блок) *прямоугольник*, содержащий имя и номер и используемый для описания функции.

Примеры таких описаний:

производить детали	планировать ресурсы	наблюдать
наблюдать за выполнением	проектировать систему	эксплуатировать
разработать детальные чертежи	изготовить компонент	проверять деталь

Интерфейсные дуги (*дуги, стрелки*) - *направленные линии*, состоящие из одного или нескольких сегментов, которые моделируют открытые каналы или каналы, передающие данные или материальные объекты от источника (*на-чальная точка стрелки*), к потребителю (*конечная точка с «наконечником»*).

Имеется 4 класса стрелок: входная стрелка, выходная стрелка, управляющая стрелка, стрелка механизма (включает стрелку вызова).

Приметы интерфейсных дуг:

Спецификации	Отчет об испытаниях	Бюджет
Конструкторские требования	Конструкция детали	Директива
Инженер-конструктор	Плата в сборе	Требования

На рисунке 3.2 приведен пример контекстной диаграммы модели IDEF0.

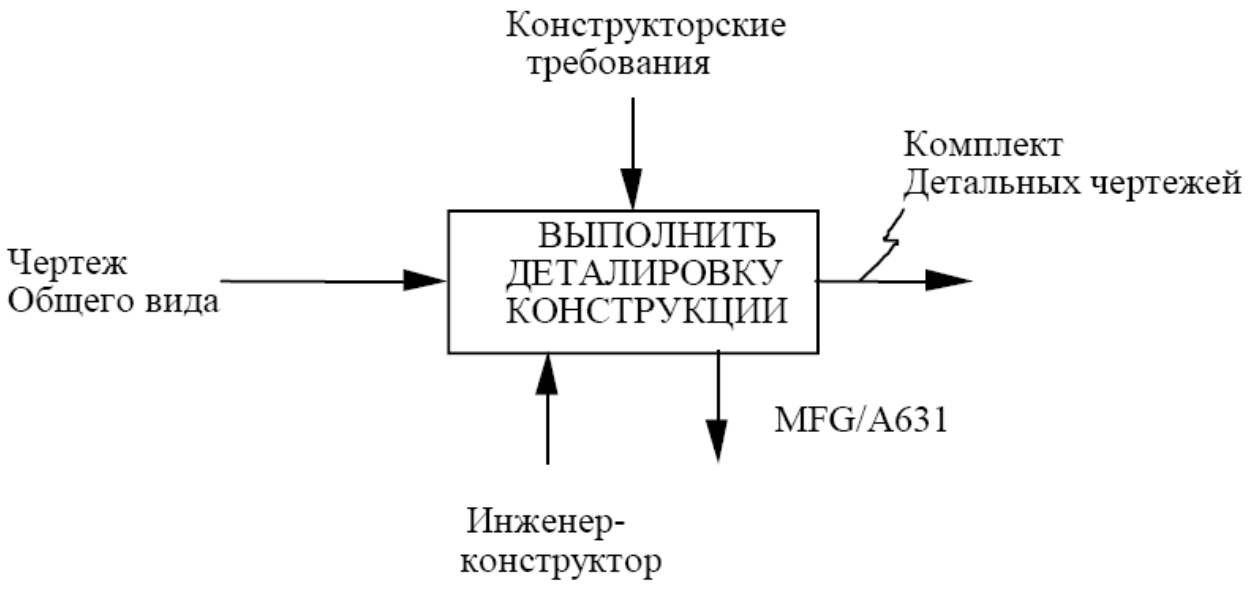

Рисунок 3.2 — Пример контекстной диаграммы

Замечание

Интерфейсные дуги моделируют информационные и/или материальные объекты.

Можно выделить *пять основных видов объектов*:

- *материальные потоки* (детали, товары, сырье и другие);
- *финансовые потоки* (наличные, безналичные, инфестиции и другие);
- *потоки документов* (коммерческие, финансовые, организационные);
- *потоки информации* (информация, данные о намерениях, устные распоряжения и другие);
- *ресурсы* (сотрудники, станки, машины и другие).

Входящими и **исходящими** интерфейсными дугами - могут отображаться все виды объектов.

Управляющими стрелками – могут быть только потоки документов и информации.

Дугами механизмов – могут быть только ресурсы.

Дополнительно, модель IDEF0 содержит:

- **Декомпозиции** – разделение моделируемой функции на функции компоненты.
- **Глосарий** - список определений для ключевых слов, фраз и аббревиатур, связанных с узлами, блоками, стрелками или с моделью IDEF0 в целом.

3.1.5 Компьютерные инструментальные средства функционального описания и исследования систем управления

Имеется множество программных продуктов, реализующих методологию IDEF0:

- *Business Studio — ГК «Современные технологии управления»;*
- *CA Erwin Process Modeler* (ранее *BPwin*) - разработчик *Computer Associates*;
- *Corel iGrafx (IDEF0)* – разработчик Corel;
- *Design/IDEF 3.5* - Фирма-разработчик: MetaSoftware (США);
- *Casewise Corporate Modeler Suit* - британской компания Casewise;
- *IDEF0\Doctor* – авторский проект Сергея Рубцова на технологии .NET;
- *IDEF0 on WEB* – страница интерфейса: http://e-a-m.ru/rsvIDEF0net.aspx
- *IDEF0.EM Tool* - Software Informer;
- *MS Visio* - Microsoft
- *Ramus* – проект RAMUS (использует Java) — коммерческий продукт; *Ramus Educational* — ограниченный бесплатный вариант *для обучения*.

Замечание

Для целей обучения по данной дисциплине, используется бесплатный вариант системы проектирования: *Ramus Educational*, установленный в среде ОС УПК АСУ.

3.2 Практические занятия №5 и №6: Создание контекстной диаграммы модели IDEF0

Тематическое назначение практического занятия **№5**: «*Создание контестной диаграммы проекта*».

Контекстная диаграмма проекта создается на основе:

- теоретического представления функциональной модели IDEF0, описанного в подразделе 3.1;
- аналитического описания системы, выполненного на занятиях №3 и №4.

Контекстная диаграмма проекта выполняется в черновом варианте: карандашом и на бумажном носителе.

Особое внимание следует уделить следующим аспектам выполнения контекстной диаграммы:

- *имеется единственный блок*, который функционально отображает *всю систему*;
- *название блока* форулируется «*глаголом в активной форме*»;
- *дуги (стрелки)* отображают информационные и/или материальные объекты, которые находятся *вне проектируой системы*.

Тематическое назначение практического занятия **№6**: «*Согласование контестной диаграммы проекта с преподавателем*».

3.3 Синтаксис и семантика графического языка IDEF0

Учебный материал данного подраздела основан на текстах документов ГОСТ РД IDEF0-2000 и ГОСТ Р 50.1.028-2001.

3.3.1 Синтаксис языка IDEF0

Блок - *описывает функцию*, основные элементы которой показаны на рисунке 3.3.
Внутри каждого блока помещаются его *имя* и *номер*.
Имя должно быть *активным глаголом* или *глагольным оборотом*, описывающим функцию.
Номер блока размещается в правом нижнем углу.
Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

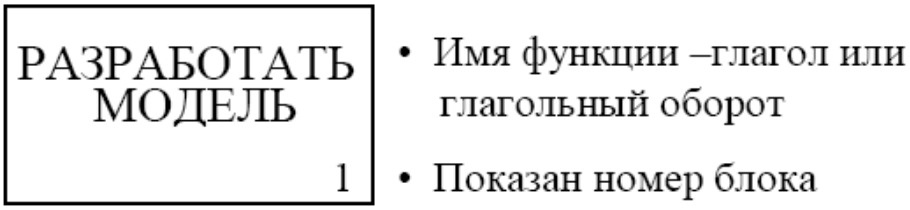

Рисунок 3.3 — Основные элементы блока

Для блоков установлены следующие *синтаксические правила*:

- *размеры блоков* должны быть достаточными для того, чтобы включить имя и номер блока.
- *блоки должны быть прямоугольными*, с прямыми углами;
- блоки должны быть *нарисованы сплошными линиями*.

Стрелка - формируется из *одного* или *нескольких отрезков прямых* и *наконечника* на одном конце.
Сегменты стрелок могут быть прямыми или ломаными; в последнем случае горизонтальные и вертикальные отрезки стрелки сопрягаются дугами, имеющими угол 90°.
Стрелки показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться и какие материальные объекты являются результатом функции.

Для стрелок установлены следующие *синтаксические правила*:

- *ломаные стрелки* изменяют направление только под углом 90°;
- стрелки должны быть нарисованы *сплошными линиями*; можно использовать линии различной толщины;

- стрелки могут состоять только из *вертикальных* или *горизонтальных отрезков*; отрезки, направленные по диагонали, не допускаются;
- *концы стрелок* должны касаться внешней границы функционального блока, но не должны пересекать ее;
- стрелки *должны присоединяться к блоку на его сторонах*; присоединение в углах не допускается.

3.3.2 Семантика языка IDEF0

Блоки именуются глаголами или глагольными оборотами, и эти имена сохраняются при декомпозиции.
Стрелки и их сегменты, как отдельные, так и *связанные в «пучок»*, помечаются существительными или оборотами существительного.
Метки сегментов позволяют конкретизировать данные или материальные объекты, передаваемые этими сегментами, с соблюдением синтаксиса ветвлений и слияний.

Общие семантические правила:
а) *Имя блока* должно быть глаголом или глагольным оборотом.
б) *Каждая сторона* функционального блока имеет стандартное отношение блок/стрелки:

- *входные стрелки* должны связываться с левой стороной блока;
- *управляющие стрелки* должны связываться с верхней стороной блока;
- *выходные стрелки* должны связываться с правой стороной блока;
- *стрелки механизма* (кроме стрелок вызова) должны указывать вверх и подключаться к нижней стороне блока;
- *стрелки вызова механизма* должны указывать вниз, подключаться к нижней стороне блока, и помечаться ссылкой на вызываемый блок.

в) *Сегменты стрелок*, за исключением стрелок вызова, должны помечаться существительным или оборотом существительного, если только единственная метка стрелки не относится к стрелке в целом.
г) *Чтобы связать* стрелку с меткой, следует использовать *ломаную молниеобразную выносную линию* ( ).

д) *В метках стрелок не должны использоваться* следующие термины: ***функция, вход, управление, выход, механизм, вызов***.

IDEF0-модели состоят из *документов трех типов*:

- графических диаграмм;
- текста;
- глоссария.

Эти документы должны иметь *перекрестные ссылки* друг на друга.

Графическая диаграмма — *главный компонент IDEF0-модели*, содержащий *блоки, стрелки, соединения блоков и стрелок и ассоциированные с ними отношения*.
Блоки представляют *основные функции моделируемого объекта*.

Эти функции:

- *могут быть разбиты* (*декомпозированы*) на составные части и представлены в виде более подробных диаграмм;
- *процесс декомпозиции* продолжается до тех пор, пока объект не будет описан на уровне детализации, *необходимом для достижения целей конкретного проекта*.

Диаграмма верхнего уровня (*родительская диаграмма*) обеспечивает наиболее общее описание объекта моделирования.
За родительской диаграммой следует серия *дочерних диаграмм*, дающих более детальное представление об объекте.

Каждая модель должна иметь *контекстную диаграмму верхнего уровня*, на которой объект моделирования представлен единственным блоком с граничными стрелками.
Эта диаграмма называется *А-0 (А минус ноль)*.
Стрелки на этой диаграмме отображают *связи объекта моделирования с окружающей средой.*

Замечание

Поскольку единственный блок представляет весь объект, его *имя - общее для всего проекта*.
Это же справедливо и для всех стрелок диаграммы, поскольку они представляют полный комплект внешних интерфейсов объекта.

Диаграмма А-0 устанавливает *область моделирования и ее границу*.
Контекстная диаграмма А-0 также должна содержать краткие утверждения, определяющие:

- *точку зрения должностного лица или подразделения*, с позиций которого создается модель;
- *цель*, для достижения которой ее разрабатывают.

Точка зрения определяет, *что и в каком разрезе* можно увидеть в пределах контекста модели.
Изменение точки зрения приводит к рассмотрению других аспектов объекта.
Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения на тот же самый объект.

Формулировка цели выражает *причину создания модели*, то есть содержит перечень вопросов, на которые должна отвечать модель, что в значительной мере определяет ее структуру.

Пример контекстной диаграммы представлен на рисунке 3.4.

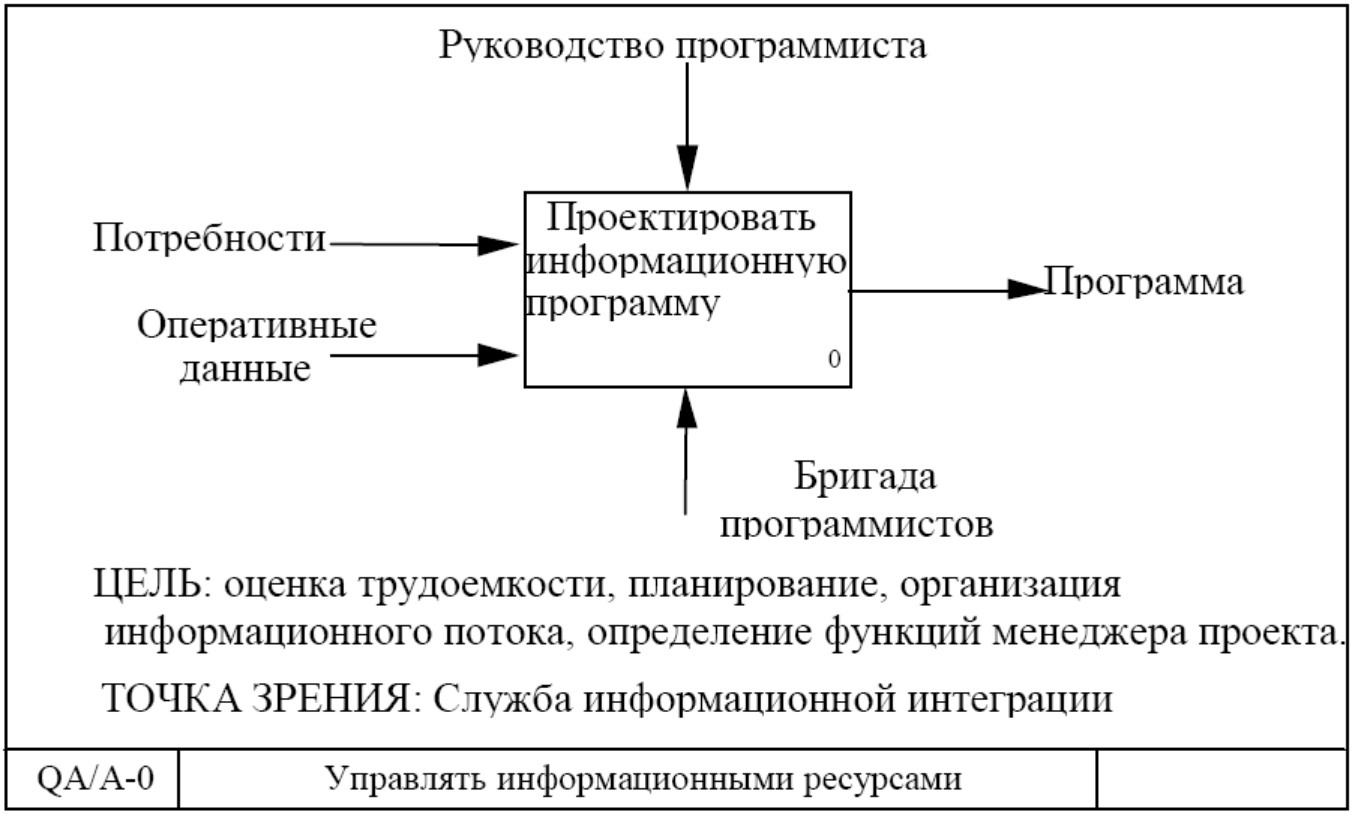

Рисунок 3.4 — Пример контекстной диаграммы

Замечание

На рисунке 3.4 имеется ошибка: надпись внизу должна совпадать с именем функции в блоке.

Наиболее важные свойства объекта обычно *выявляются на верхних уровнях иерархии*; по мере декомпозиции функции верхнего уровня и разбиения ее на подфункции, эти свойства уточняются.

Каждая подфункция, в свою очередь, декомпозируется на элементы следующего уровня, и так происходит до тех пор, *пока не будет получена релевантная структура*, позволяющая ответить на вопросы, сформулированные в цели моделирования.

Каждая подфункция моделируется отдельным блоком.

Каждый родительский блок подробно описывается дочерней диаграммой на более низком уровне.

Все дочерние диаграммы должны быть в пределах области контекстной диаграммы верхнего уровня.

Единственная функция, представленная на контекстной диаграмме верхнего уровня, *может быть разложена на основные подфункции* посредством создания ***дочерней диаграммы***.

Каждая из подфункций может быть разложена на составные части посредством создания дочерней диаграммы более низкого уровня.

Некоторые или все функции дочерней диаграммы также могут быть разложены на

составные части.

Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.
Дочерняя диаграмма, создаваемая при декомпозиции, охватывает ту же область, что и родительский блок, но описывает ее более подробно.
На рисунке 4.5 представлен эскизный пример построения дочерних диаграмм.

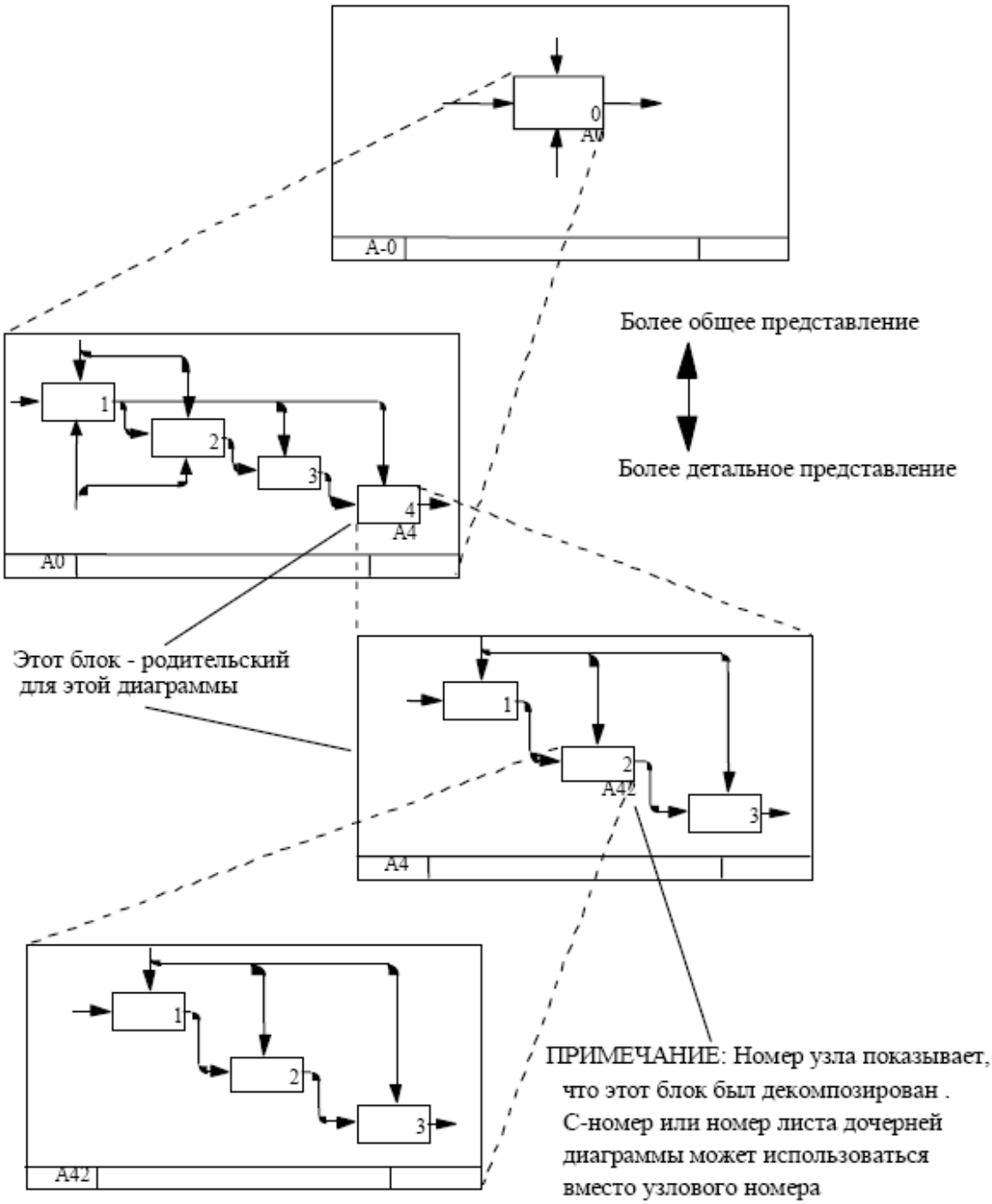

Рисунок 4.5 — Примеры декомпозиции

Часто, для нумерации создаваемых диаграмм используется ***С-номер***.

С-номер - *номер, создаваемый в хронологическом порядке* для идентификации диаграммы и прослеживания ее истории; он может быть использован в качестве ссылочного выражения при определении конкретной версии диаграммы.
Обычно С-номер состоит из инициалов автора модели и хронологических данных (даты создания очередной версии диаграммы, то есть — внутренний номер системы разработки).

Правила кодирования самих блоков представлены на рисунке 3.6:

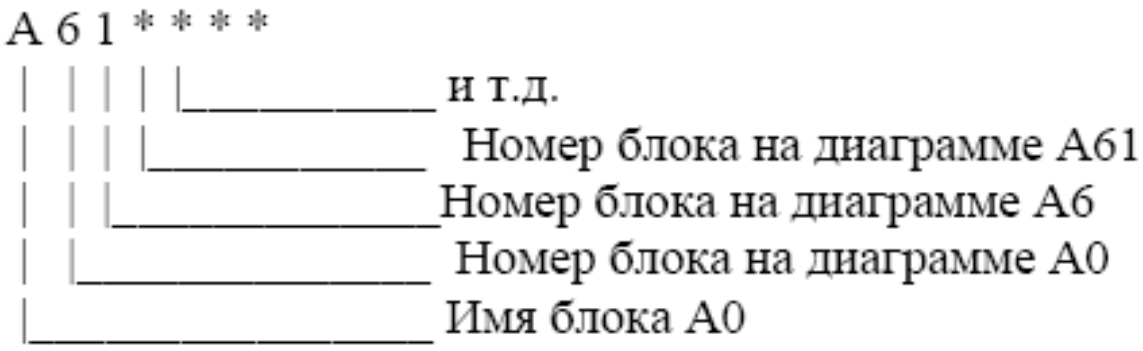

Рисунок 3.6 — Правила кодирования блоков

Диаграмме *может быть поставлен* в соответствие *структурированный текст*, представляющий собой краткий комментарий к содержанию диаграммы.
Текст используется для объяснений и уточнений характеристик, потоков, внутриблочных соединений и других целей.
Текст не должен использоваться для описания и без того понятных блоков и стрелок на диаграммах.
При большом объеме текст располагается на отдельном листе модели (текстовой странице).

Глоссарий предназначен для определения аббревиатур, ключевых слов и фраз, используемых в качестве имен и меток на диаграммах.
Глоссарий определяет понятия и термины, которые должны быть одинаково понимаемы всеми участниками разработки и пользователями модели, чтобы правильно интерпретировать ее содержание.
Глоссарий составляется с любой необходимой степенью подробности.

Диаграммам могут быть сопоставлены *диаграммы-иллюстрации (**FEO** - For Exposition Only Diagrams)*, которые используются в качестве дополнений, поясняющих специфику содержания основных диаграмм в случаях, когда это необходимо.
Диаграмма FEO не должна подчиняться синтаксическим правилам IDEF0.
Допускается параллельное выполнение блоков, что помечается метками на стрелках.
Допускается ветвление и слияние стрелок.

3.4 Практическое занятие №7: «ЦЕЛЬ» и «ТОЧКА ЗРЕНИЯ» проекта

Тематическое назначение практического занятия **№7**: «*Правильное определение понятий «ЦЕЛЬ» и «ТОЧКА ЗРЕНИЯ» проекта*».
Предполагается, что студент изучил теоретический материал подраздела 3.3.

Важность данной темы занятия обусловлена тем, что обычно:

- студент *неправлиьно понимает* термин «ЦЕЛЬ», интерпретируя его как имя функции контекстной диаграммы;
- студент *неправлиьно понимает* термин «ТОЧКА ЗРЕНИЯ», интерпретируя его как собственную точку зрения или как точку зрения одного из «*механизмов*» системы.

Следует помнить, что:

- **ЦЕЛЬ** — это обозначение понятия, содержащего информацию о назначении проекта, обычно, сформулированное **Заказчиком** проекта; например: «Для определения функциональной струкуры системы»; «По заказу министерства обороны ...»; «Для демонстрации на научной конференции ...»;
- **ТОЧКА ЗРЕНИЯ** - лицо, группа лиц или организация, определяющие направленность восприятия проекта и форулирующие требования к нему; обычно — это руководящее лицо проектируеой системы.

Важно заметить, что указанные понятия воздействуют на проект как по раздельности, так и в совместной совокупности.

Например, *точки зрения на офис* строителя, сантехника и дизайнера — *различны*, а *на политику государства* - *могут совпадать*.

Военные обычно создают системы, приносящие максимальные материальные разрушения.
Строители - создают атериальные ценности.
Философы и **религия** создают духовные ценности.

Таким образом, *«ЦЕЛЬ»* и *«ТОЧКА ЗРЕНИЯ» проекта* оказывают существенное влияние на весь проект и, как следствие, - изменяют декомпозицию его функций.

Замечание

«ЦЕЛЬ» и «ТОЧКА ЗРЕНИЯ» обязательно должны быть индивидульно *согласованы с преподавателем!*

3.5 Отношения блоков на диаграммах IDEF0

В методологии IDEF0 существует *шесть типов отношений* между блоками в пределах одной диаграммы:

- доминирование;
- управление;
- выход — вход;
- обратная связь по управлению;
- обратная связь по входу;
- выход — механизм.

3.5.1 Отношения доминирования

Доминирование определяется взаимным расположением блоков на диаграмме. Предполагается, что блоки, расположенные на диаграмме *выше и левее*, *«доминируют»* над блоками, расположенными *ниже и правее*.
«**Доминирование**» понимается как влияние, которое один блок оказывает на другие блоки диаграммы, что показано на рисунке 3.7.

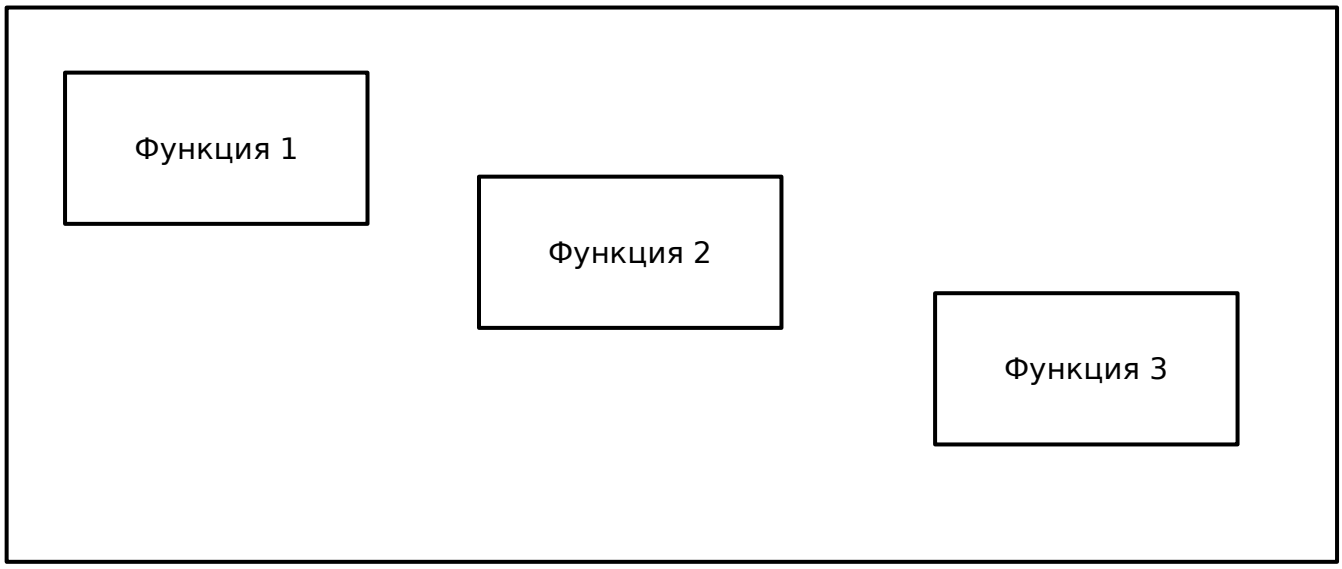

Рисунок 3.7 — Отношения доминирования

3.5.2 Отношение управления

Отношение управления, показанное на рисунке 3.8, возникает, когда *выход одного* блока служит *управляющим воздействием* на *блок с меньшим доминированием*.

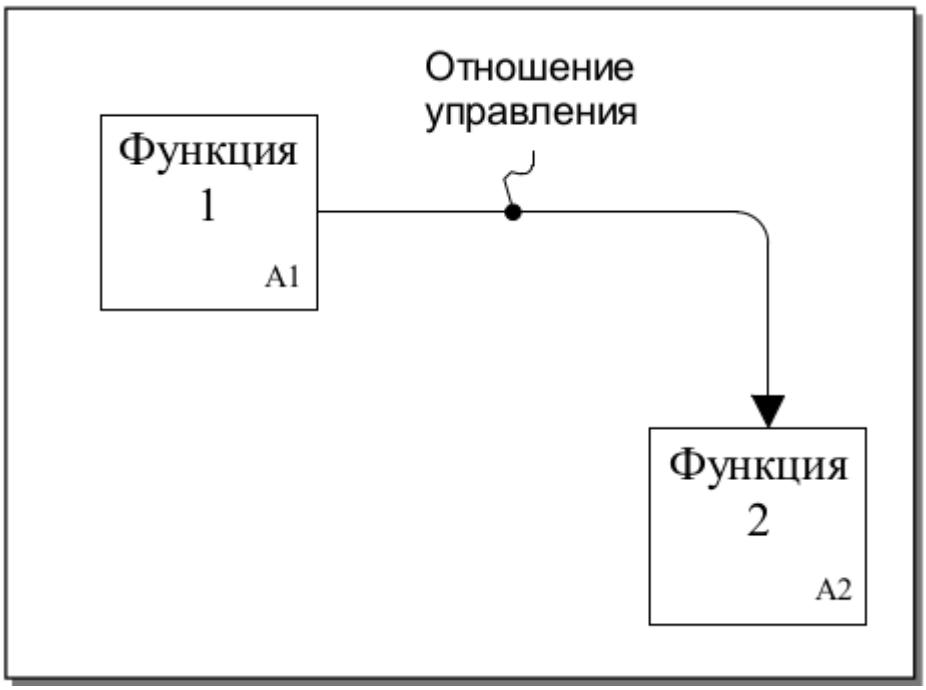

Рисунок 3.8 — Отношение управления

3.5.3 Отношение выход-вход

Отношение выход — вход, показанное на рисунке 3.9, возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием.

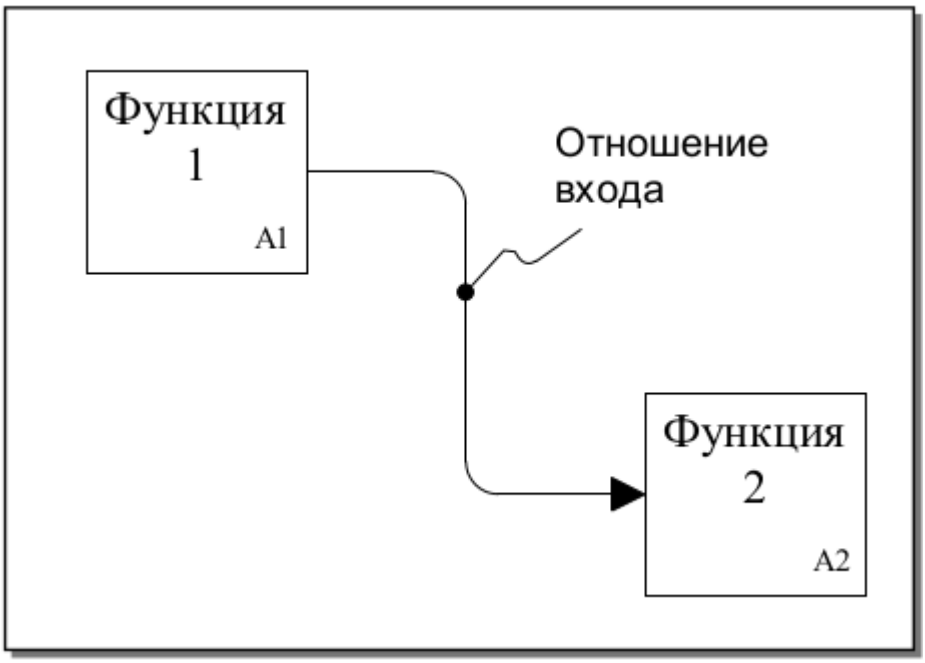

Рисунок 3.9 — Отношение выход-вход

3.5.4 Обратная связь по управлению

Обратная связь по управлению, как показано на рисунке 3.9, возникает, когда *выход* некоторого блока создает *управляющее воздействие* на *блок с большим доминированием*.

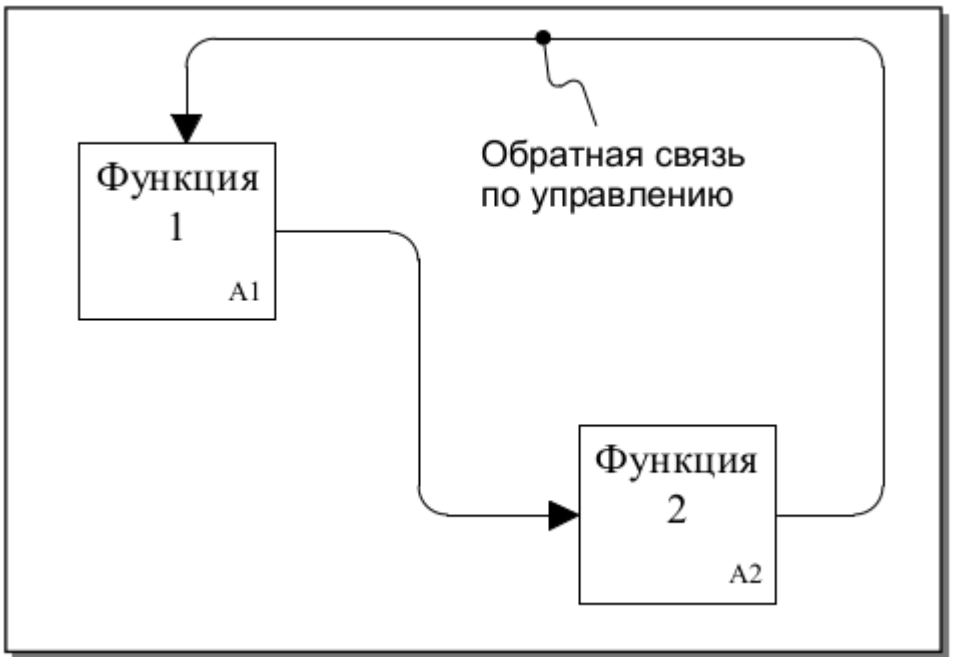

Рисунок 3.9 — Отношение обратной связи по управлению

3.5.5 Обратная связь по входу

Отношение обратной связи по входу имеет место, как показано на рисунке 3.10, когда *выход блока становится входом* другого блока с большим доминированием.

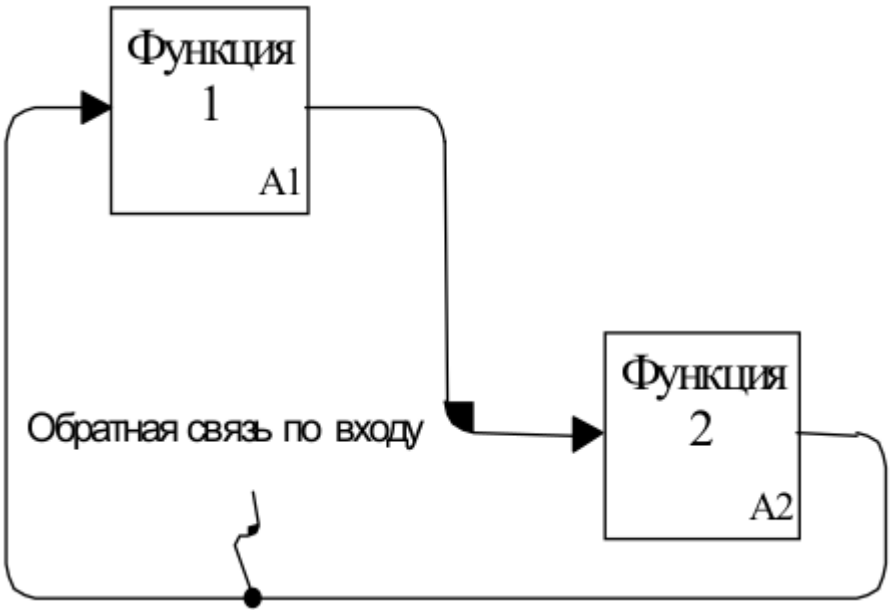

Рисунок 3.10 — Отношение обратной связи по входу

Замечание

Обратная связь по управлению и обратная связь по входу являются более сложными типами отношений, поскольку они представляют итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием, что впоследствии влияет на исходную функцию).

3.5.6 Связь «вход-механизм»

Связь «выход — механизм», показанная на рисунке 3.11, отражает ситуацию, при которой выход одной функции становится средством достижения цели для другой функции.

Связи «выход — механизм» возникают при отображении в модели процедур пополнения и распределения ресурсов, создания или их подготовки.

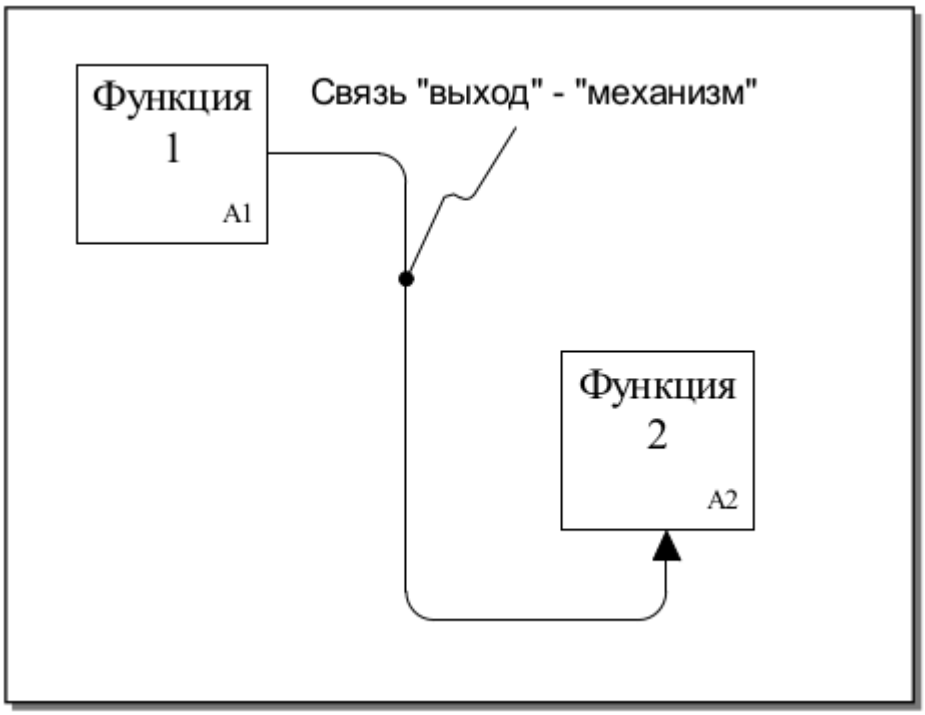

Рисунок 3.11 — Связь «выход-механизм»

Таким образом, перечисленный набор из шести отношений блоков, графически отображает особенности связей функций, которые должны быть учтены в процессе проектирования системы — на этапах декомпозиции блоков диаграмм модели IDEF0.

3.6 Уточняющие элементы диаграмм IDEF0

Процесс декомпозиции модели системы порождает множество объектов и связей. **С целью** обеспечения должного уровня надежности и обеспечения контроля всего проекта в него вводятся четыре основных уточняющих элемента:

- ICOM-кодирование;
- стрелки, помещенные в тунель;
- перечень узлов;
- дерево узлов.

3.6.1 ICOM-кодирование граничных стрелок

ICOM-коды связывают **граничные стрелки на дочерней диаграмме** со стрелками родительского блока, как показано на рисунке 3.12.

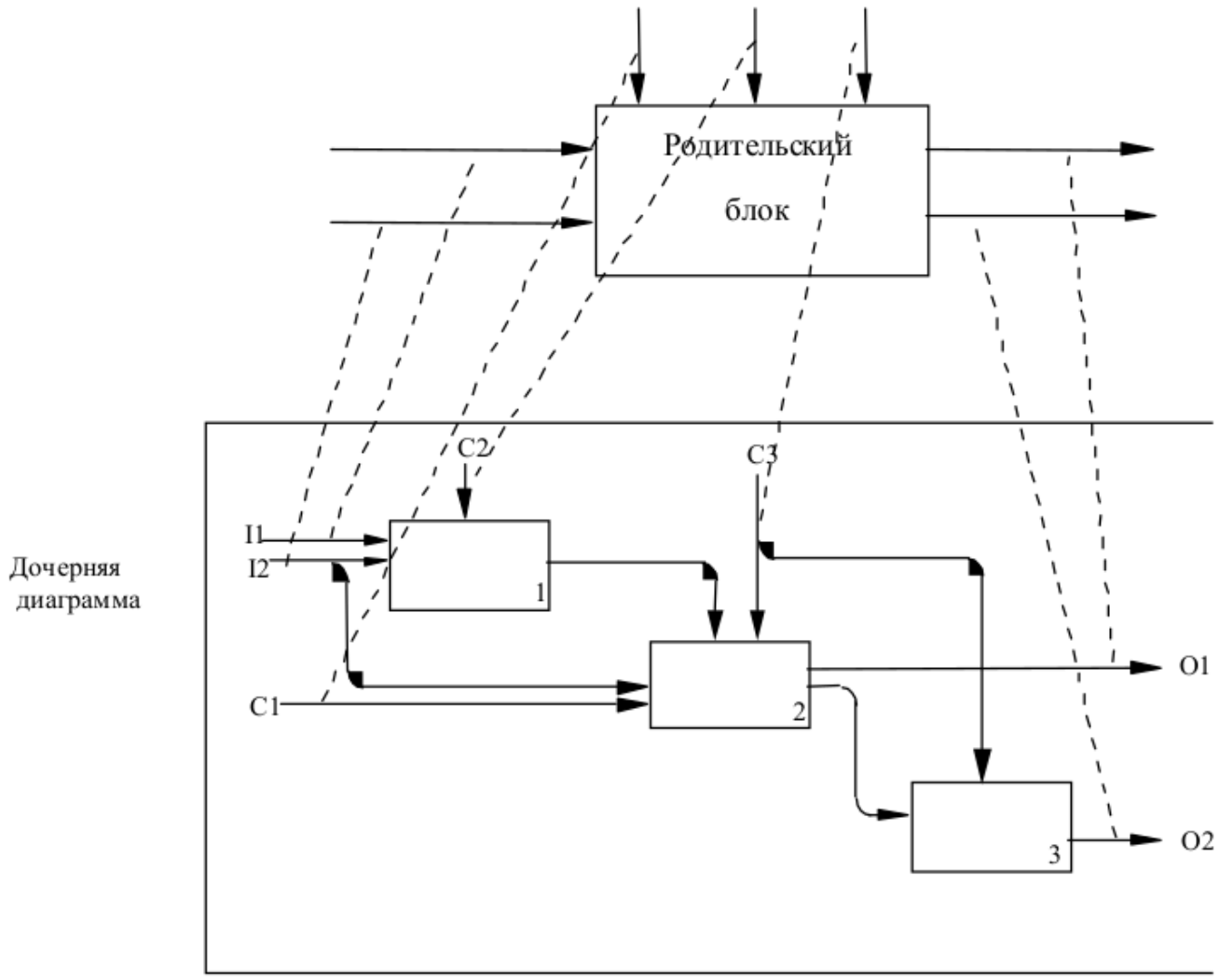

Рисунок 3.12 — Пример ICOM-кодирования стрелок

Буквы *I, C, O* или *M*, приведенные около несвязанного конца граничной стрелки на дочерней диаграмме, идентифицируют стрелку как *Вход* (Input), *Управление* (Control), *Выход* (Output) или *Механизм* (Mechanism) в родительском блоке.

Число следующее за буквой, определяет относительное положение точки подключения стрелки к родительскому блоку; это положение определяется слева направо или сверху вниз.
Например, код «С3», показанный на рисунке 3.12, возле граничной стрелки на дочерней диаграмме указывает, что эта стрелка соответствует третьей (считая слева) управляющей стрелке родительского блока.
Это кодирование связывает каждую дочернюю диаграмму со своим родительским блоком.
Если блоки на дочерней диаграмме подвергаются дальнейшей декомпозиции и подробно описываются на дочерних диаграммах следующего уровня, то *на каждую новую диаграмму назначаются новые ICOM-коды*, связывающие граничные стрелки этих диаграмм со стрелками их родительских блоков.

Замечание
Иногда буквенные ICOM-коды, определяющие роли граничных стрелок (*вход, управление, механизм*), могут меняться при переходе от родительского блока к дочерней диаграмме.
Например, управляющая стрелка на родительском блоке может быть входом на дочерней диаграмме.
Иногда, вход родительского блока может быть управлением для одного или нескольких дочерних блоков.

3.6.2 Стрелки, помещенные в «туннель»

Туннельные стрелки, как показано на рисунках 3.13 и 3.14, означают, что данные, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме.

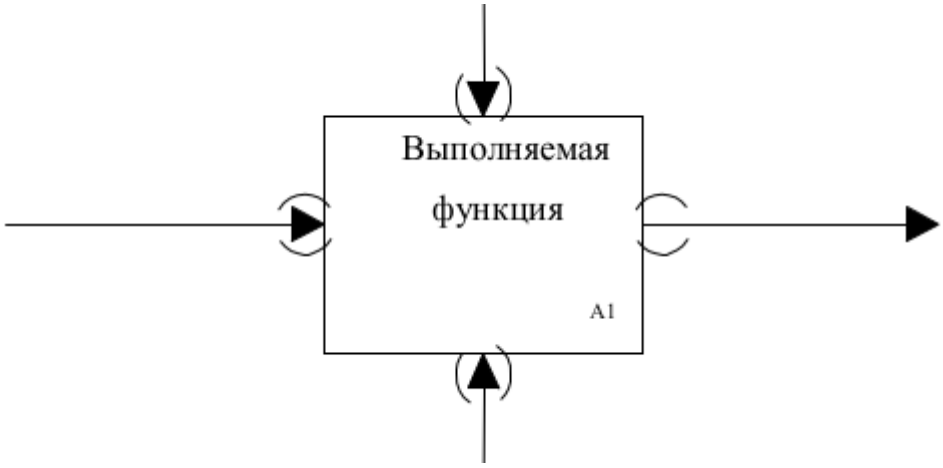

Рисунок 3.13 — Указанные стрелки в блоке не рассматриваются

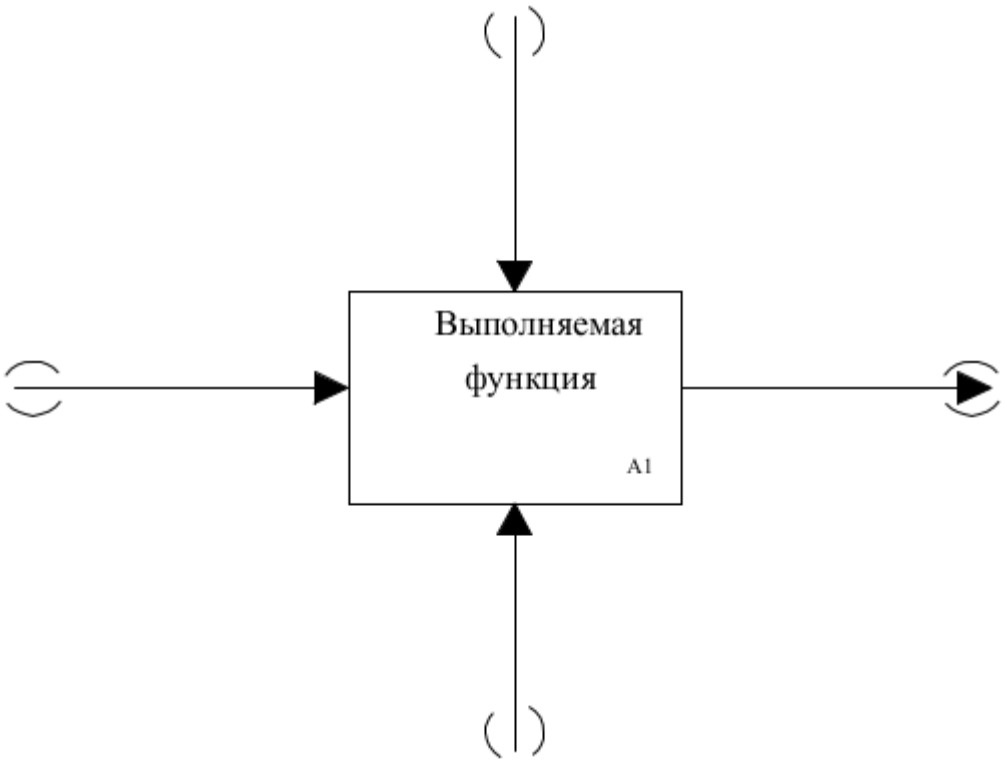

Рисунок 3.14 — Указанные стрелки на родительской диаграмме не рассматриваются

Туннель обозначается круглыми скобками в начале и/или в конце стрелки.

Стрелка, помещенная в туннель *там, где она присоединяется к блоку*, как показано на рисунке 3.13, означает, что данные, выраженные этой стрелкой, *не обязательны на следующем уровне декомпозиции*.

Стрелка, помещаемая в туннель *на свободном конце*, как показано на рисунке 3.14, означает, что представляемые ею данные *отсутствуют на родительской диаграмме*.

3.6.3 Перечень узлов

Узел – это функциональный блок, порождающий дочерние блоки.
Перечень узлов представляет информацию о входящих в модель узлах в форме списка, наподобие обычного оглавления и отражает ее иерархическую структуру.

Структура перечня узлов — список строк:

- содержащих номер узла (блока) и его название;
- имеющих отступы от начала строки, соответсвующие уровню диаграммы на которой они расположены.

Пример перечня узлов:

A0 Производить продукцию
- A1 Планировать производство
  - A11 Выбрать технологию производства
  - A12 Оценить требуемое время и затраты на производство
  - A13 Разработать производственные планы
  - A14 Разработать план вспомогательных действий
- A2 Планировать и управлять графиком выпуска и ресурсами
  - A21 Разработать основной график
  - A22 Разработать график координации работ
  - A23 Оценивать затраты и приобретать ресурсы
  - A24 Следить за выполнением графика и расходом ресурсов
- A3 Планировать выпуск продукции

3.6.4 Дерево узлов

Разработанная модель IDEF0, со всеми уровнями структурной декомпози-цией может быть представлена на единственной диаграмме *в виде дерева уз-лов*, дополняющего перечень узлов.
Для изображения этого дерева нет стандартного формата.
Единственное требование, как показано на рисунке 3.15, состоит в том, что вся иерархия узлов модели *должна быть представлена наглядно и понятно*.

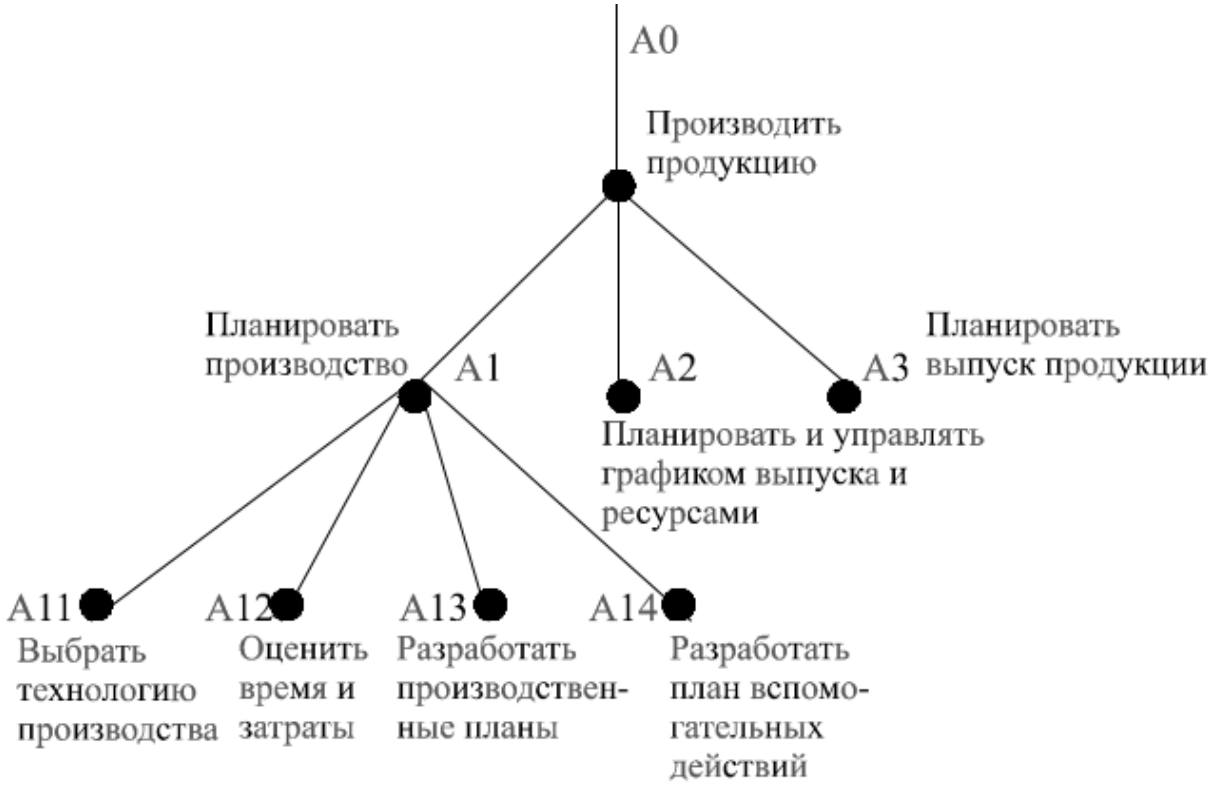

Рисунок 3.15 — Пример дерева узлов

3.7 Практическое занятие №8: Декомпозиция модели проекта

Тематическое назначение практического занятия **№8**: «*Проведение декомпозиции диаграмм проекта*».

Декомпозиция модели проводится на основе:

- уже утвержденной на занятии №7 контекстной диаграммы;
- с учетом отношений блоков на диаграммах, описанных в подразделе 3.5;
- с использованием уточняющих элементов диаграмм, описанных в подразделе 3.6.

Исполнение диаграммы проводится на бумажном носителе и карандашем, с учетом возможной последующей коррекции и уточнения.

Минимальные требования к декомпозиции:

- *диаграмма А0*, соответствующая декомпозиции контекстной диаграммы А-0;
- *декомпозиция любого блока*, выбранного на диаграмме А0;
- *наличие всех надписей* (имен) как на блоках, так и дугах (стрелках).

Рекомендации:

- *количество блоков* на диаграммах от 3 до 5;
- *функциональная нагруженность* блоков диаграммы должна быть максимально равномерной.

Замечание

Проводимая декомпозиция (построение) диаграмм преследует учебные цели.
В общем случае:

- требования стандарта IDEF0 не ограничивают глубину декомпозиции;
- декомпозиция заканчивается, когда функциональные блоки диаграмм определяют *элементарные действия*, не требующие дальнейшей детализации;
- понятия элементарного действия формулируются заказчиком проекта.

Результат декомпозиции, в обязательном порядке, должен пройти *согласование с преподавателем!*

3.8 Методические приемы разработки моделей IDEF0

ГОСТ РД IDEF0-2000 и **ГОСТ Р 50.1.028-2001** содержат описания методики функционального анализа, применительно к *системам уровня предпреиятия*.
Соответственно, учебный материал данного подраздела следует указанной методике, предполагая интерпретацию его к системам автоматизированного управления (АСУ).

3.8.1 Обобщенные определения

Материальный поток - непрерывное или дискретное множество материальных объектов, распределенное во времени.
Информационный поток - множество информационных объектов, распределенное во времени.
Информация, участвующая в процессах, операциях, действиях и деятельности в целом, как показано на рисунке 3.16, может быть *классифицирована на три группы*:

- ограничительная;
- описательная;
- предписывающая (управляющая).

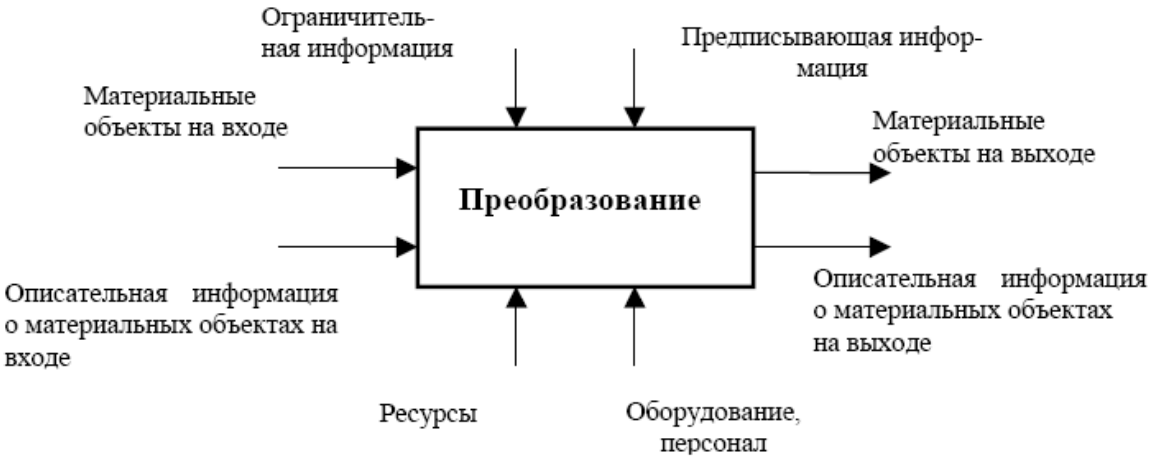

Рисунок 3.16 — Три группы информации на диаграмме IDEF0

3.8.2 Классификация функций

В практике построения функциональных моделей желательно иметь некоторую классификацию явлений и событий, отображаемых в моделях.
Такая классификация облегчает:

- *выбор глубины декомпозиции* моделируемых систем;
- *способствует выработке* единообразных подходов и приемов моделирования в конкретных предметных областях.

3.8.3 Четыре основных вида функций

Деятельность (синонимы: *дело, бизнес*) - совокупность **процессов**, выполняемых (протекающих) последовательно или/и параллельно, преобразующих множество материальных или/и информационных потоков во множество материальных или/и информационных потоков с другими свойствами**.**
Она осуществляется в соответствии с заранее определенной и постоянно корректируемой **целью**, с потреблением финансовых, энергетических, трудовых и материальных ресурсов, *при выполнении ограничений со стороны внешней среды*.
В модели IDEF0 деятельность определяется блоком *A0* на основной контекстной диаграмме *A-0*.
При моделировании крупных, многопрофильных структур (фирм, организаций, предприятий), которые по своему статусу занимаются различными видами деятельности, последние представляют собой различные *экземпляры класса «деятельность»* и могут найти отражение в дополнительной контекстной диаграмме ***A-1***.
В этом случае, общая модель сложной структуры будет состоять из ряда частных моделей, каждая из которых относится к конкретному виду деятельности.

Процесс (синоним: *бизнес-процесс*) - совокупность последовательно или/и параллельно выполняемых операций, преобразующая материальный или/и информационный потоки в соответствующие потоки с другими свойствами.
*Он (**Процесс**)* протекает в соответствии с управляющими директивами, вырабатываемыми на основе ***целей деятельности***.
В ходе процесса потребляются финансовые, энергетические, трудовые и материальные ресурсы и выполняются ***ограничения*** со стороны других процессов и внешней среды.

Операция - совокупность последовательно или/и параллельно выполняемых ***действий***, преобразующих объекты, входящие в состав материального или/и информационного потока, в соответствующие объекты с другими свойствами.

Операция выполняется:

- в соответствии с ***директивами***, вырабатываемыми на основе директив, определяющих протекание процесса, в состав которого входит операция;
- с потреблением всех видов необходимых ***ресурсов***;
- с соблюдением ***ограничений*** со стороны других операций и внешней среды.

Действие - преобразование какого-либо свойства материального или информационного объекта *в другое свойство*.
Действие выполняется в соответствии с **командой**, являющейся частью **директивы** на выполнение операции, с потреблением необходимых ресурсов и с соблюдением ограничений, налагаемых на осуществление операции.

3.8.4 Дополнительные виды функций

Субдеятельность - *совокупность нескольких процессов* в составе деятельности, объединенная некоторой частной целью, являющейся «подцелью» деятельности.
Подпроцесс - *группа операций в составе процесса*, объединенная технологически или организационно.
Уровни декомпозиции, *детализирующие действия*, естественно считать состоящими из *элементарных* или *простых функций*.

3.8.5 Организационно-технические структуры и механизмы

Выделенные классы функций образуют *иерархическую подчиненность* по принципу «*сверху вниз*»:

- деятельность;
- субдеятельность;
- процесс;
- подпроцесс
- операция;
- действие.

Согласно методологии IDEF0 каждая функция выполняется посредством некоторого механизма.

Во многих системах такими механизмами служат *организационно-технические структуры*.
Соответственно, можно выделить *иерархическую подчиненность механизмов*, как показано на рисунке 3.17.

Рисунок 3.17 — Иерархическая подчиненность механизмов

Организационно-техническая система - организационная структура, персонал и комплекс технических средств (оборудование), необходимые для осуществления деятельности.

Организационно-техническая подсистема - часть организационно-технической системы, обеспечивающая протекание процесса (субдеятельности).

Организационно-технический комплекс (модуль) - часть организационно-технической подсистемы, предназначенная для выполнения операции.

Организационно-технический блок — часть организационно-технического комплекса, обеспечивающая выполнение действия.

3.8.6 Управление

Методология IDEF0 – методология *описания и проектирования систем управления*.
Следуя методике иерархической подчиненности функций, можно привести некоторую иерархию управляющих объектов и действий с этими управляющими объектами.

Управление деятельностью - ***процесс***, состоящий как минимум из следующих **операций**:

- *формулирование целей* деятельности;
- *оценивание ресурсов*, необходимых для осуществления деятельности и их сопоставление с имеющимися ресурсами;
- *сбор информации* об условиях протекания и фактическом состоянии деятельности (*«глобальная» обратная связь*);
- *выработка и принятие решений*, направленных на достижение целей, в частности, решений о распределении ресурсов по процессам, входящим в состав деятельности; оформление решений в виде директив на управление процессами;
- *реализация решений* (исполнение директив) и оценка их результатов (*«локальная обратная связь»*);
- *корректировка*, в случае необходимости, например при нехватке ресурсов, ранее сформулированных целей (*самонастройка, адаптация*).

Управление процессом - ***операция***, состоящая как минимум из следующих действий:

- *анализ директивы* на управление процессом, ее декомпозиция на директивы управления операциями;
- *сбор (прием по каналам связи) информации* о ходе выполнения операций, ее обобщение и формирование сведений о состоянии процесса;
- *передача данных* в подсистему управления деятельностью;
- *сопоставление информации* о ходе операций с данными директив и выработ-

ка локальных решений, направленных на устранение отклонений;
- *корректировка*, в случае необходимости, директив на выполнение операций.

Управление операцией - ***действие***, состоящее в выработке на основании директивы на управление операцией команд на управление действиями, в реализации этих команд, оценке результатов выполнения, передаче необходимой информации в комплекс управления процессом, корректировке команд в случае необходимости.

Управление действием – как правило, это - *элементарная функция*.

Замечание

На IDEF0-диаграммах могут быть отдельные **функциональные блоки**, определяющие преобразование **управляющих объектов**.
Через них осуществляются управляющие воздействия на остальные блоки диаграммы.
Именно эти блоки воспринимают ограничивающую и предписывающую информацию и преобразуют ее в соответствующие директивы и команды.
Имена блоков управления, как правило, содержат глагол «**Управлять . . .** ».

3.9 Практическое занятие №9: Завершение декомпозиции модели

Тематическое назначение практического занятия **№9**: «*Завершение декомпозиции диаграмм проекта*».

Завершение декомпозиции модели проводится с учетом теоретического материала, изложенного в подразделе 3.8.

Главное назначение данного практического занятия:
- *привести названия* функциональных блоков в соответствие с методикой, изложенной в подразделе 3.8;
- *согласовать* все структурные изменения и связи, существующие в модели, для последующей документации ее в автоматизированной среде проектирования.

Процесс согласования модели проекта проводится:
- *индивидуально*, - по каждому проекту;
- *публично*, - посредством доклада на практическом занятии.

Замечание

Считается, что на данном занятии учебная цель, связанная с творческой частью процесса проектирования заканчивается.
Дальнейшие практические занятия ориентированы на изучение инструментальной части, предназначенной для обеспечения автоматизации проектного процесса.

3.10 Состав участников проекта и их взаимодействие

IDEF0-модель есть *результат скоординированной коллективной работы*, как показано на рисунке 3.18, при которой:

- *авторы создают* первоначальные диаграммы, основанные на собранной информации об объекте моделирования;
- *авторы передают* их другим участникам проекта для рассмотрения и формулирования замечаний.

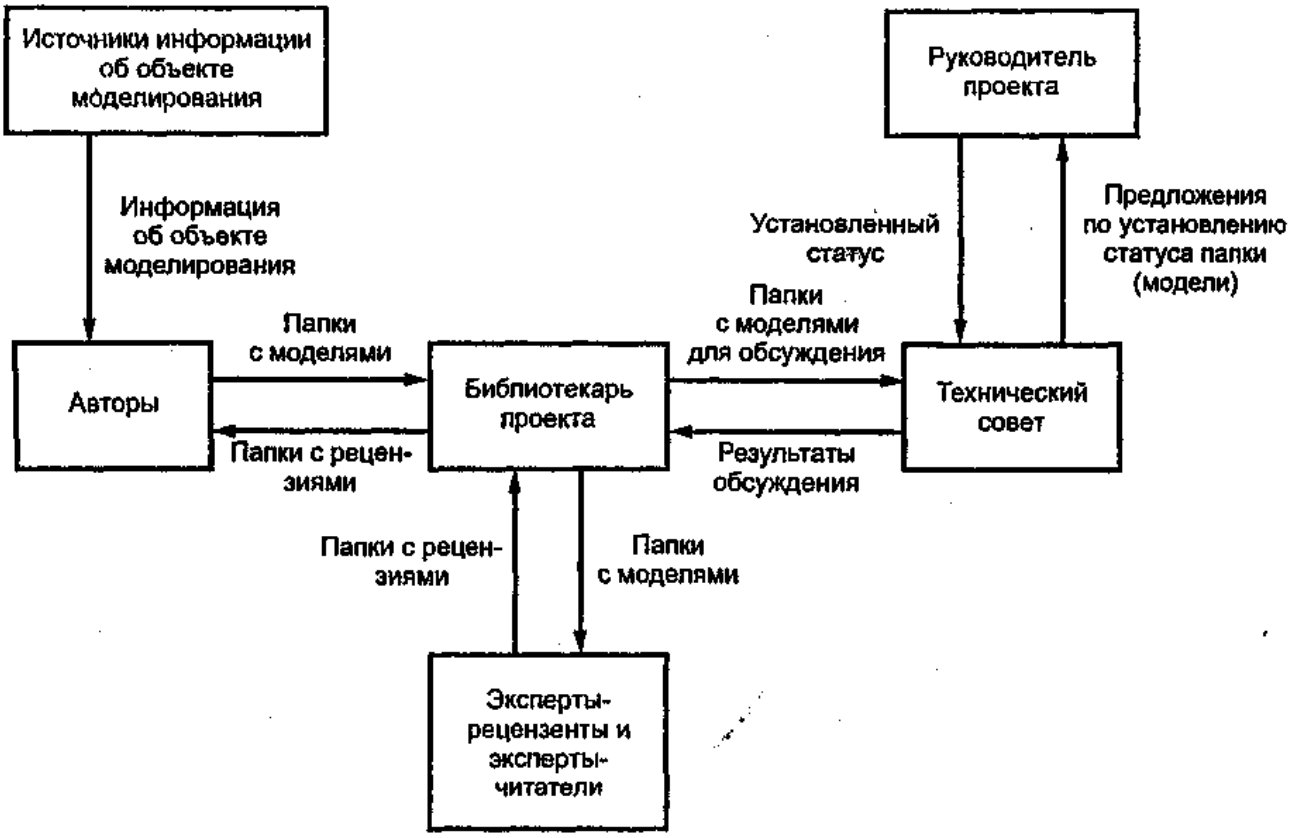

Рисунок 3.18 — Состав участников проектной группы

Руководитель проекта - должностное лицо, осуществляющее административное управление проектом;

Разработчик проекта - лицо, создающие IDEF0-модель на основе материала, собранного из источников информации;

Технический совет - элемент организации процесса создания моделей, предлагающий арбитражные решения по моделированию и рекомендации по установлению статуса диаграмм, части и/или модели в целом;

Эксперт - выбираемое руководителем проекта лицо, обладающее специальными знаниями некоторых аспектов моделируемой области;

Библиотекарь - лицо, ответственное за хранение документации, изготовление копий, координацию обмена письменной и/или электронной информацией: *рассылка папок, получение рецензий, регистрация и публикация диаграмм и модели*;

Источники информации - *Исходная информация* для IDEF0-модели поступает к разработчику из разных источников: *от людей и от документов*.

Замечание

Каждый эксперт, у которого есть замечания к диаграмме, должен сделать их письменно и передать автору диаграммы.
В общем случае, *процесс моделирования* является итеративным и должен привести к точному описанию системы.

3.11 Практическое занятие №10: Рабочее место ОС УПК АСУ

Тематическое назначение практического занятия **№10**: «*Изучение инструментальных средств ОС УПК АСУ, в которой располагается среда проектирования*».

Учебный материал данного практического занятия изложен в учебно-методическом пособии [3].

В процессе занятия, студент должен освоить:

- *процесс запуска* ОС УПК АСУ в учебных классах кафедры АСУ или на личных компьютерах;
- *средства конфигурирования* программного обеспечения GRUB2, обеспечивающего загрузку ОС.

3.12 Практическое занятие №11: Настройка среды ОС УПК АСУ

Тематическое назначение практического занятия **№11**: «*Освоить среду обучения, касающейся изучаемой дисциплины*».

Учебный материал данного практического занятия изложен в учебно-методическом пособии [3].
Архив учебной рабочей области, по данной дисциплине, *студент обязан получить у преподавателя!*

В процессе занятия, студент должен освоить:

- *процесс поключения архива* к ОС УПК АСУ;
- *инфрастуктуру учебной среды* по дисциплине «*Автоматизация проктирования систем и средств управления*»;
- *процедуру архивации* личной среды и копирование ее на личный flashUSB.

3.13 Стандартный бланк методологии IDEF0

Стандартный бланк (Мастер-страница, шаблон), показанный на рисунке 3.19, обычно имеет формат А4 и содержит три области:

- **Область проектной информации** — в верхней части страницы.
- **Рабочая область**, в которой располагается диаграмма или другая информация — в центральной части страницы.
- **Область идентификации** — вдоль нижнего края страницы.

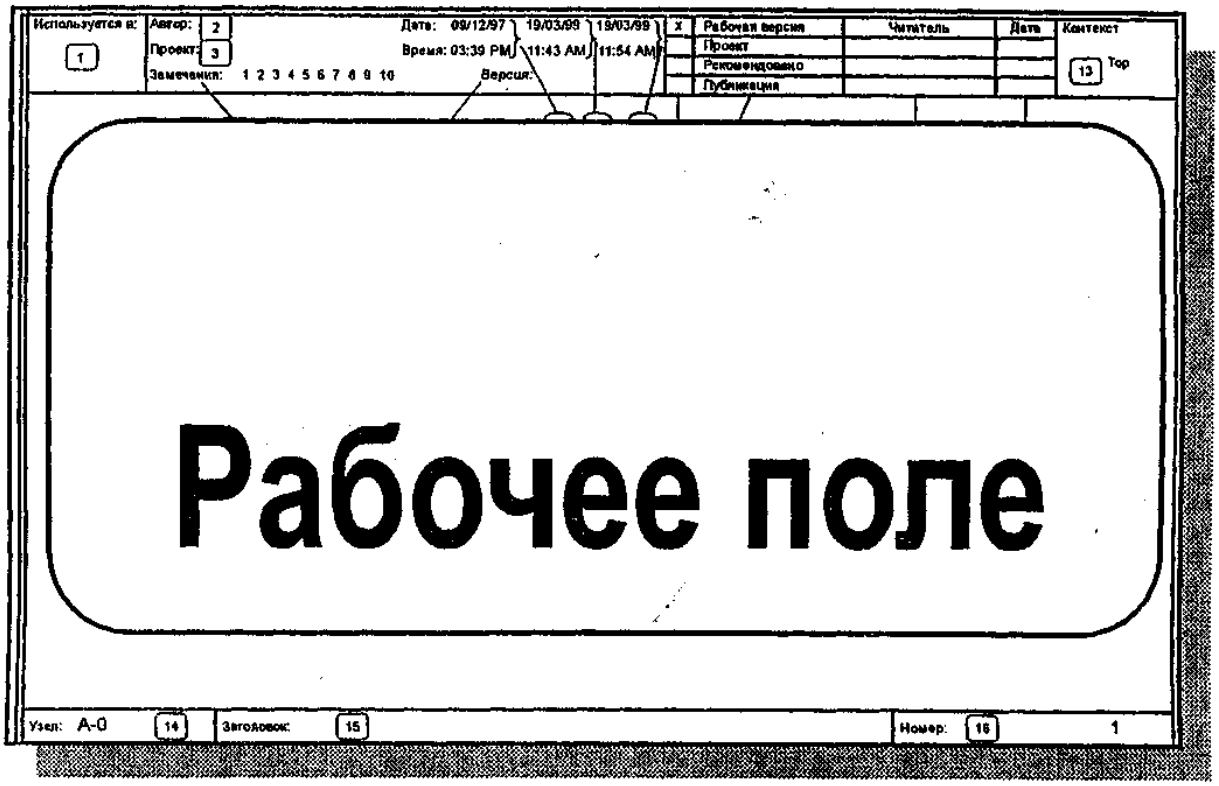

Рисунок 3.19 — Стандартный бланк модели IDEF0

3.13.1 Область проектной информации

1 Поле «Используется в:»

Содержит название общей модели, составной частью которой является настоящая модель, или область ее применения.

2 Поле «Автор»

В этом поле указывают имя автора диаграммы или наименование организации— разработчика проекта.

3 Поле «Проект»

Содержит название проекта или его аббревиатуру.

4 Поле «Замечания»

Используется для указания количества замечаний на текущей странице. Чтобы указать количество замечаний, соответствующее число перечеркивают крестом (символом «X») или обводят это число эллипсом.

5 Поле «Версия»

Используется для указания информации о версии диаграммы.

6 Дата и время создания страницы.

7 Дата и время внесения последнего изменения на странице.

8 Текущая дата и время.

9 Поля для установления статуса страницы.

В зависимости от присвоенного статуса, в соответствующем поле ставится символ «X».

10 Наименования устанавливаемых статусов.

Наименования статуса, присваиваемого диаграмме по мере ее продвижения по циклам «автор — чтитель». Приняты следующие виды статусов диаграмм в зависимости от уровня их готовности.

«**Рабочая версия**» - диаграмма с большим числом изменений. Новые диаграммы являются рабочими версиями.

«**Проект**» - диаграмма имеет меньше изменений по сравнению с предыдущим уровнем, что свидетельствует о достижении некоторого согласия с экспертами.

Статус «Проект» присваивается диаграмме по предложению руководителя работ, до ее рассмотрения экспертами и одобрения техническим советом.

«**Рекомендовано**» - диаграмма и сопутствующие ей тексты прорецензированы и утверждены техническим советом. В нее не предполагается внесение существенных изменений.

«**Публикация**» - материал, который может быть представлен заказчику модели и рассылаться для окончательной печати и опубликования.

11 Поля для подписей читателей (экспертов) диаграммы.

12 Дата прочтения диаграммы экспертом.

13 Поле «Контекст»

Содержит миниатюрную картинку страницы диаграммы (без стрелок), содержащей родительский блок, который на этой картинке затемнен. Это позволяет иметь перед глазами структуру родительской диаграммы и место на ней декомпозируемого блока, не вызывая диаграмму на экран.

3.13.2 Область идентификации

14 Поле «Узел»
В поле помещается **номер родительского блока** для этой страницы диаграммы.

15 Поле «Заголовок»
В поле помещается **текст имени родительского блока**, в результате декомпозиции которого создана текущая страница.

16 Поле «Номер»
Поле содержит номер данной страницы, на который имеется ссылка на родительской диаграмме (под декомпозируемым блоком).
Это может быть С-номер или номер страницы.

3.14 Практическое занятие №12: Изучение среды разработки RAMUS

Тематическое назначение практического занятия **№12**: «*Освоить систему автоматизированного проектирования RAMUS*».

Учебная цель — освоение запуска и индивидуальной настройки системы RAMUS. Работа проводится на компьютерах учебных классов кафедры АСУ или персональных компьютерах студентов.

3.15 Практическое занятие №13: Создание проекта в среде RAMUS

Тематическое назначение практического занятия **№13**: «*Освоить в инструментальной среде RAMUS процедуры создания и сохранения проектируемых систем*».

Учебная цель:
- *освоить запуск* автоматизированной среды разработки RAMUS и создание проекта для модели IDEF0;
- *реализация* контекстной диаграммы проекта в среде разработки RAMUS;
- *начало формирования* электронного вариана отчета по проекту в среде ОС УПК АСУ.

Контролируемый результат работы студента:
- *уверенная работа* в среде ОС УПК АСУ и системе проектирования RAMUS;
- *наличие шаблона отчета* по проектируемой системе.

3.16 Практическое занятие №14: Создание макета отчета по проекту

Тематическое назначение практического занятия **№14**: «*Реализация в инструментальной среде RAMUS полного проекта системы*».

Учебная цель:

- *создание полного проекта* в автоматизированной среде разработки RAMUS;
- *формирование электронного вариана отчета* по всему проекту.

Контролируемый результат работы студента:

- *наличие проекта* в среде системы проектирования RAMUS;
- *наличие отчета* по проектируемой системе.

3.17 Функциональное моделирование в методике IDEF3

В данном подразделе изложены краткие сведения о дополнительной модели функционального моделирования, известной как IDEF3.

Методология IDEF3 (*Integrated Definition Process Description Capture Method*) была разработана *с целью более удобного описания рабочих процессов* (***Work Flow Definition***).

WFD используется для процессов, в которых *важно отразить логическую последовательность выполнения процедур*.
На рисунке 3.20 представлен пример модели IDEF3.

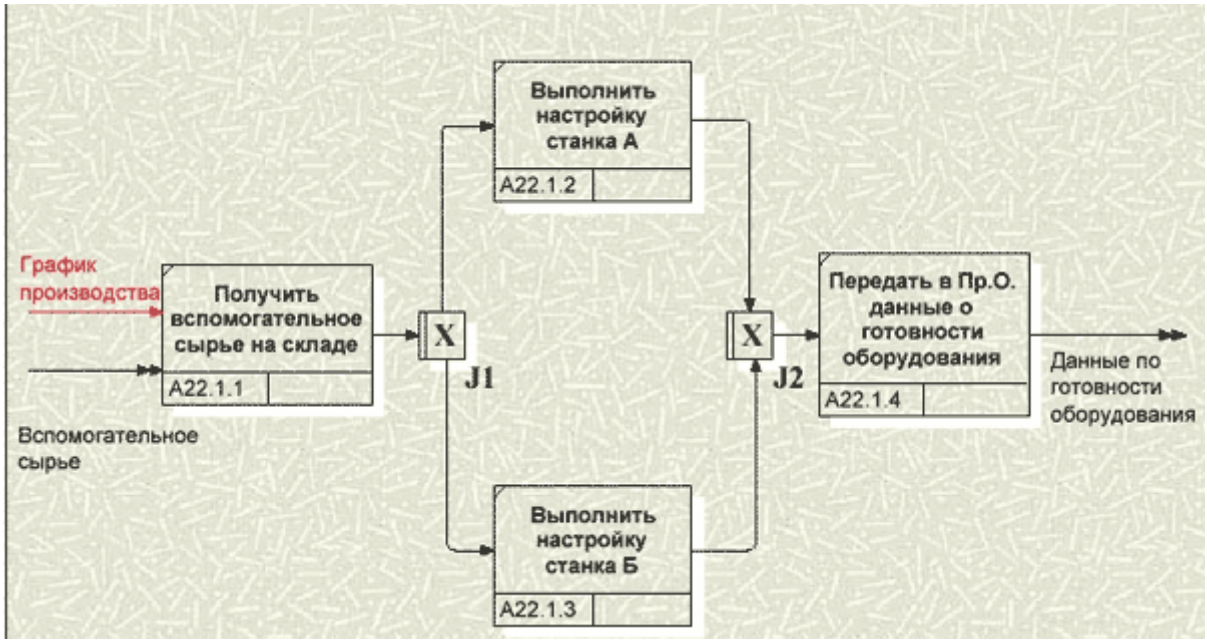

Рисунок 3.20 — Пример модели IDEF3 (WFD)

Замечание

Эта методика, в отличие от IDEF0, *плохо стандартизирована*.
С ее описанием можно познакомиться на сайте http://www.idef.com.

IDEF3 - это структурный метод, который показывает:

- *причинно-следственные* связи и события,
- *как организована работа*, и какие пользователи работают с моделируемой системой.

IDEF3 состоит из *двух методов*:

- *Process Flow Description (PFD)* - описание процессов, с указанием того, *как организована работа* между различными элементами моделируемой системы;
- *Object State Transition Description (OSTD)* - *описание переходов состояний объектов*, с отображением промежуточных состояний у объектов в моделируемой системе.

С помощью IDEF3 описываются **сценарий** и **последовательность** операций для каждого процесса.

Сценарием называется *описание последовательности изменения свойств объекта* в рамках рассматриваемого процесса.
Например, описание последовательности этапов обработки детали в цеху и изменение ее свойств после прохождения каждого этапа.

Исполнение каждого сценария сопровождается *соответствующим документооборотом*, который состоит из двух потоков:

- документы, определяющие *структуру и последовательность процесса* (технологические указания, описания стандартов);
- документы, отображающие *ход его выполнения* (результаты экспертиз, отчеты о браке).

Средства документирования и моделирования IDEF3 позволяют выполнять следующие задачи:

- *документировать имеющиеся данные* о технологии процесса;
- *определять и анализировать точки влияния потоков* сопутствующего документооборота на сценарий технологических процессов;
- *определять ситуации*, в которых требуется принятие решения, влияющего на жизненный цикл процесса (например, изменение технологических свойств конечного продукта);
- *содействовать принятию оптимальных решений* при реорганизации технологических процессов;
- *разрабатывать имитационные модели* технологических процессов по принципу «как будет, если...».

IDEF3 имеет прямую взаимосвязь с методологией IDEF0 - каждая функция может быть представлена в виде отдельного процесса средствами IDEF3.
Отличие IDEF3 от моделирования в IDEF0 и DFD в том, что IDEF3 *отражает функции системы во временной последовательности их осуществления*.

3.17.1 Синтаксис IDEF3

Компоненты синтаксиса языка IDEF3 – это: *единицы работы, стрелки, перекрестки, объекты ссылок, диаграммы, правила*.

Единица работы описывает процесс, действие, решение.

Стрелки показывают последовательность выполнения работ.

Перекрестки используются в диаграммах, чтобы:

- *показать ветвления логической схемы* моделируемого процесса;
- *альтернативные пути развития процесса*, которые могут возникнуть во время его выполнения.

Правила определяют: как следует применять компоненты.

Диаграммы обеспечивают формат графического и словесного описания модели.
На рисунке 3.21 показаны элементы диаграммы, используемые IDEF3.

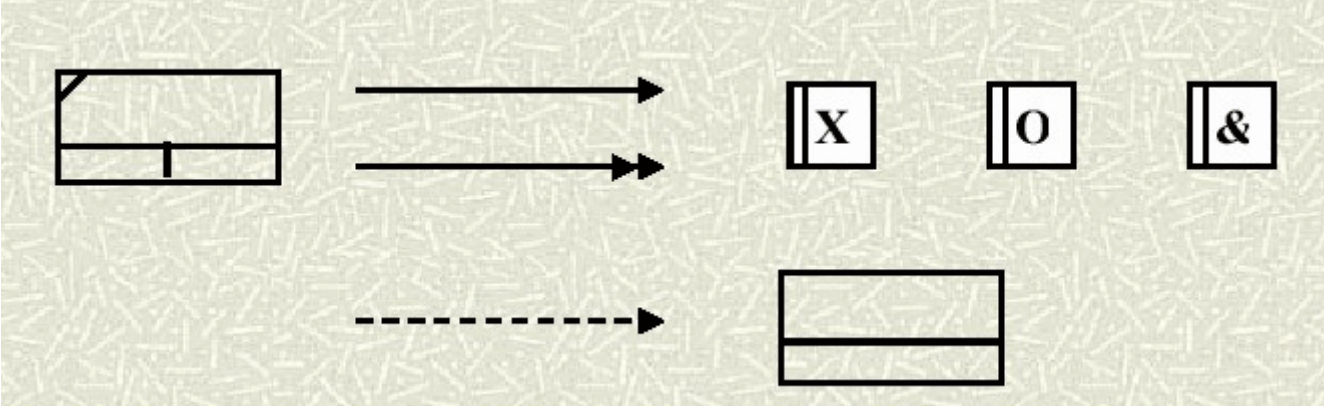

Рисунок 3.21 — Элементы даграмм модели IDEF3

Важное значение на диаграммах имеют *перекрестки* (*Junction*).
Перекрестки используются для *отображения логики взаимодействия стре-лок* (потоков).

Все перекрестки нумеруются, каждый номер имеет *префикс "J"*.

Классификация возможных типов перекрестков приведена в таблице 3.1.

Таблица 3.1 — Логические значения перекрестков

Обозначение	***Наименование***	***Случай слияния стрелок (Fan-in Junction)***	***Случай разветвления стрелок (Fan-out Junction)***
&	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
&	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
O	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
O	Synchronous OR	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно
X	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Перекресток *не может использоваться одновременно* для слияния и для разветвления.
При внесении перекрестка в диаграмму необходимо указать тип перекрестка.

3.17.2 Семантика IDEF3

Единица работы - это основной компонент диаграммы IDEF3, близкий по смыслу к блоку IDEF0.
Он изображается прямоугольником с прямыми углами, с именем и номером.

Работа *именуются* отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы.
Другое имя существительное, в составе той же фразы, обычно изображает основной результат работы: например, приготовление фарша.
Номер единицы работы присваивается при ее создании и *не меняется никогда*. Даже если работа будет удалена, ее номер не будет вновь использоваться.
Обычно, номер состоит из номера родительской работы и порядкового номера на текущей диаграмме.

Работа требует более подробного описания, чем блок в IDEF0.

Каждая единица работы должна иметь документ, который включает текстовое описание компонентов работы:

- *объектов и фактов*, связанных с ней;
- *ограничений*, накладываемых на работу;
- *дополнительное описание* работы.

Стрелки на диаграмме IDEF0 означают потоки информации или объектов, передаваемые от одной функции к другой.
На диаграмме IDEF3 стрелки могут показывать *только последовательность выполнения работ*, то есть имеют иной смысл, нежели стрелки IDEF0.

Различают три типа связей изображаемых стрелками:

- *связь предшествования;*
- *поток объектов;*
- *связь отношения.*

Связь предшествования показывает, что *работа-источник должна завершиться прежде, чем начнется работа-приемник*.
Такая связь обозначается **сплошной линией**.
Связь должна быть именована таким образом, чтобы была понятна причина ее появления.

Поток объектов показывает участие некоторого объекта в двух или более единицах работы.
Обозначается стрелкой с двумя наконечниками.
Наименования потоковых связей должны четко идентифицировать объект, который передается с их помощью.

Связь отношения - показывает связь между двумя работами или между работой и объектом ссылки.
Обозначается пунктирной линией.
Используются для отражения отношений между работами, которые *невозможно описать* с использованием связей предшествования или потока объектов.
Один из примеров связи отношения - *отображение взаимоотношений между параллельно выполняющимися работами*.

Связь отношения является *альтернативой другим связям* в смысле задания последовательности выполнения работ: работа-источник не обязательно должна закончиться прежде, чем работа-цель начнется.

Объекты ссылок служат для выражения идей и концепций без использования *стрелок, перекрестков или работ*.
Объекты ссылок используются *для привлечения внимания пользователей* к каким-либо важным аспектам модели.

Перекрестки используются в диаграммах IDEF3, чтобы показать ветвления логи-

ческой схемы моделируемого процесса и альтернативные пути развития процесса, могущие возникнуть во время его выполнения.
В отличии от IDEF0, в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Различают перекрестки для слияния и разветвления стрелок:

- *Перекресток слияния* - узел, собирающий множество стрелок в одну, указывая на необходимость условия завершенности работ-источников стрелок для продолжения процесса.
- *Перекресток ветвления* - узел, в котором единственная входящая в него стрелка ветвится, показывая, что работы, следующие за перекрестком, выполняются параллельно или альтернативно.

3.18 Методология DFD

Методология DFD (Data Flow Diagrams) - *диаграммы потоков данных*.
Это - способ представления процессов обработки информации, который подобно IDEF0, *представляет систему как сеть процессов*, связанных между собой с помощью стрелок.

Авторы методики Гейн и Сарсон разработали ее независимо от IDEF0.
Эта методика, в отличие от IDEF0, *не стандартизирована*.
Пример типичной диаграммы DFD показан на рисунке 3.22.

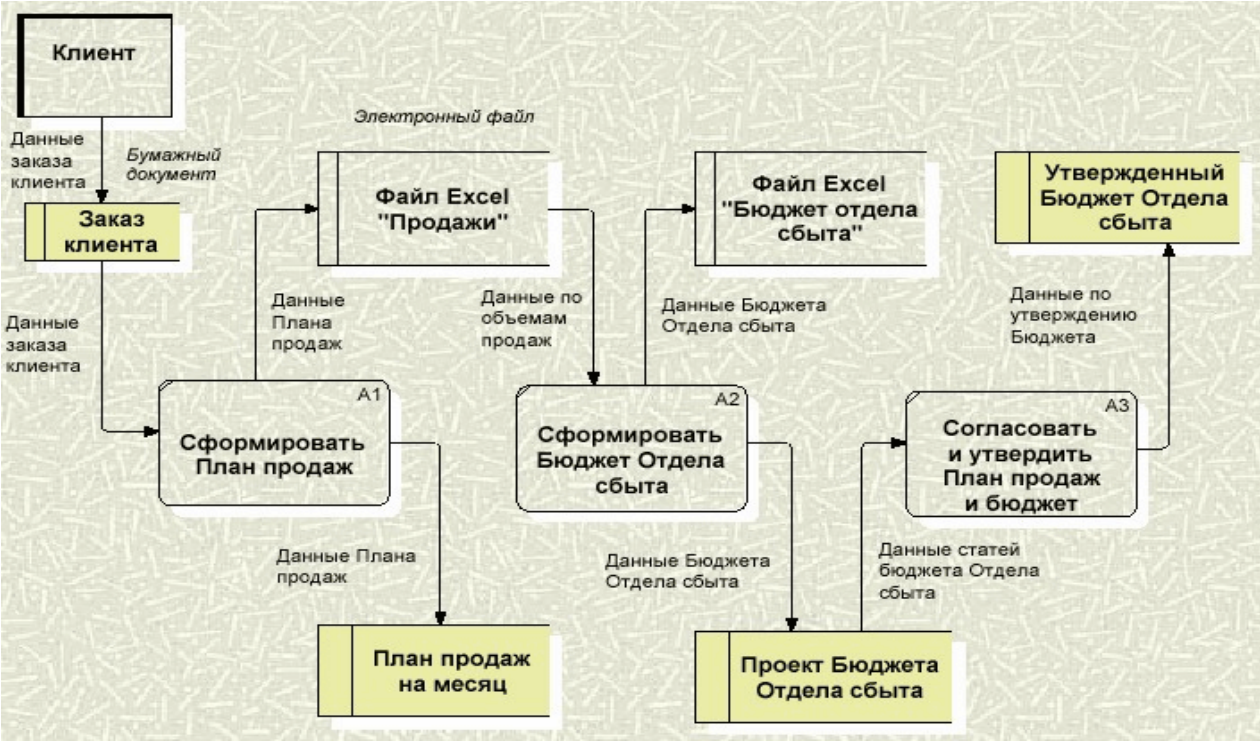

Рисунок 3.22 — Пример типичной диграммы DFD

Стрелки модели DFD (*потоки данных*) не имеют жестких связей как в IDEF0.
Они показывают, как объекты реально перемещаются от одной функции к другой.
Такое представление обеспечивает отражение таких физических характеристик системы, как:

- *движение объектов;*
- *хранение объектов;*
- *распространение объектов.*

Диаграммы DFD обеспечивают удобный способ описания передаваемой информации, как между частями моделируемой системы, так и между системой и внешним миром.
Область применения DFD: создание моделей информационного обмена организации, например, модели документооборота.

Также, DFD широко применяется при построении корпоративных информационных систем.

3.18.1 Синтаксис DFD

Синтаксис модели DFD опирается на синтаксис языка IDEF3 – это:

- *процессы,*
- *потоки данных,*
- *хранилища данных,*
- *внешние сущности,*
- *диаграммы,*
- *правила.*

Процессы представляют собой функции системы, преобразующие входы в выходы.

Потоки данных определяют информацию, передаваемую через некоторое соединение от источника к приемнику.

Хранилища данных представляет собой абстрактное устройство для хранения информации.

Внешняя сущность моделирует объект, воздействующий на систему извне.

Диаграммы обеспечивают формат графического и словесного описания модели.

Правила определяют, как следует применять компоненты.

3.18.2 Семантика DFD

Процессы представляют собой функции системы, преобразующие входы в выходы. *Процессы изображаются* прямоугольниками со скругленными углами.
Смысл процесса совпадает с семантикой блока в IDEF0 и единицами работы в IDEF3.
Процессы имеют входы и выходы, но *не поддерживают управление и меха-низмы*.

Каждый процесс *должен быть* именован глаголом с последующим дополнением и уникальный номер.
Физически процесс может быть реализован различными способами:

- *подразделение организации*, выполняющее обработку входных документов и выпуск отчетов;
- *программа*;
- аппаратно реализованное *логическое устройство*.

Потоки данных определяют *информацию, передаваемую через некоторое соединение от источника к приемнику*.

Поток данных изображается линией, заканчивающейся стрелкой, которая показывает направление потока.
Каждая стрелка имеет имя, отражающее его содержание.
Реальный поток данных может быть:

- *информацией*, передаваемой по кабелю между двумя устройствами;
- пересылаемыми по почте *письмами*;
- *магнитными дискетами*, переносимыми с одного компьютера на другой.

Каждая сторона блока не имеет четкого назначения - стрелки могут подходить и выходить из любой грани блока.
Двунаправленные стрелки служат для описания диалогов типа *команда-ответ*.

Поток данных описывает *объекты в движении*.

Хранилища данных изображают *объекты в покое*.
Хранилище представляет собой *абстрактное устройство для хранения информации*, которую можно в любой момент поместить в хранилище и через некоторое время извлечь, причем способы помещения могут быть любыми.
Имя хранилища должно идентифицировать его содержимое и быть существительным.
Физически хранилище может быть реализовано в виде ящика в картотеке, таблицы в оперативной памяти, файла на магнитном диске.
Хранилище в общем случае является *прообразом будущей базы данных*.

Внешняя сущность представляет собой материальный *объект или физическое лицо*, представляющее собой источник или приемник информации.
Например, заказчик, персонал, поставщик, клиенты, склад.
Имя сущности должно содержать *существительные*.
Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что *она находится за пределами границ анализируемой системы*.

Внешняя сущность обозначаются в виде *прямоугольника с тенью* и обычно располагаются по краям диаграммы.
Одна внешняя сущность может быть использована многократно на одной диаграмме.
Такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

3.19 Практическое занятие №15: Предварительная защита отчета

Два последних практических занятия посвящены анализу и оценке навыков, полученных студентами в результате освоения курса изучаемой дисциплины.

Тематическое назначение практического занятия **№15**: «*Предварительная защита полного проекта системы*».

Учебная цель:
- *подготовка документации* по проектирумой системе;
- *формирование навыков защиты* результатов проекирования.

Контролируемый результат работы студента:
- *наличие проекта* на проектируемую систему;
- *наличие документации* по результатам проектирования.

3.20 Практическое занятие №16: Защита проекта системы

Тематическое назначение практического занятия **№16**: «*Защита проекта системы*».

Учебная цель:
- *оценка результатов получения практических навыков*;
- *допуск к экзаменационному оцениванию*.

Контролируемый результат работы студента:
- *наличие проекта* на проектируемую систему;
- *наличие документации* по результатам проектирования;
- *результат защиты* студентом индивидуального проекта.

Заключение

Изложенный в данном пособии учебный материал предоставляет студенту методические рекомендации и теоретические представления по изучаемой дисциплине «Автоматизация проектирования систем и средств управления» уровня образовательной программы магистратура по направлению подготовки: 09.04.01 «Информатика и вычислительная техника».

Напрвлкенность учебного материала на функциональное моделирование АСУ обоснована как тематической направленностью кафедры, так и стремлением уравновесить подготовку студента, ориентированного на прикладную разработку систем управления, знаниями, дополняющими его теоретические представления, сформированные изучением технологий объектно-ориентированного программирования (ООП).

Ограниченный объем теоретической подготовки, выделенный на данную дисциплину, не позволил в полной мере осветить все аспекты функционального моделирования автоматизированных систем. Тем не менее, все вопросы связанные с построением моделей стандарта IDEF0 освещены в объеме, достаточном для самостоятельного автоматизированного проектирования.

Весь теоретический курс обучения по данной дисциплине закрепляется полным набором практических занятий, реализуемых на вычислительной инфраструктуре кафедры АСУ, что обеспечивает формирование реальных практических навыков студентов в процессе реализации индивидуальных проектов.

Список использованных источников

1 Кудрявцев Е.М. Основы автоматизированного проектирования [Текст] : учебник для вузов / Е. М. Кудрявцев. - М. : Академия, 2011. - 304 с.

2 Норенков И.П. Основы автоматизированного проектирования: Учебник для вузов. М.: МГТУ им. Н.Э. Бауана, 2002. -336 с.

3 Резник В.Г. Учебный программный комплекс кафедры АСУ на базе ОС ArchLinux. Учебно-методическое пособие. – Томск, ТУСУР, 2016. – 33 с. [Электронный ресурс]. - http://asu.tusur.ru/learning/books/b13.pdf

Учебное издание

Резник Виталий Григорьевич

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ СИСТЕМ И СРЕДСТВ УПРАВЛЕНИЯ

Практические занятия и теория функционального моделирования

Учебно-методическое пособие предназначено для изучения основ теории и проведения практических занятий по дисциплине «Автоматизация проектирования систем и средств управления» для студентов уровня образовательной программы магистратура направления подготовки: 09.04.01 «Информатика и вычислительная техника».

Учебно-методическое пособие

Усл. печ. л. 7,62. Тираж . Заказ .
Томский государственный университет
систем управления и радиоэлектроники
634050, г. Томск, пр. Ленина, 40