

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

В.Г. Резник

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебное пособие

Тема 3. Концептуальное проектирование ИС

Томск
2023

УДК 004.8 (004.9)
ББК 65.2-5-05

Резник, Виталий Григорьевич

Проектирование информационных систем. Тема 3. Концептуальное проектирование ИС. Учебное пособие / В.Г. Резник. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2023. – 67 с.

Учебное пособие предназначено для обучения дисциплине «Проектирование информационных систем» для студентов направлений подготовки бакалавриата: 09.03.01 — «Информатика и вычислительная техника» и 09.03.03 — «Прикладная информатика».

Одобрено на заседании каф. АСУ протокол №_____ от _____

УДК 004.8 (004.9)
ББК 65.2-5-05

© Резник В. Г., 2023
© Томск. гос. ун-т систем упр. и радиоэлектроники, 2023

Оглавление

3 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ ИС.....	4
3.1 Функциональное моделирование.....	5
3.1.1 Функциональное моделирование бизнес-процессов.....	5
3.1.2 Применимость функциональных моделей к стадиям проектирования ИС.....	9
3.2 Методология стандарта IDEF0.....	11
3.2.1 Стандартный состав проектной группы.....	11
3.2.2 Синтаксис и семантика языка IDEF0.....	12
3.2.3 Контекстные диаграммы IDEF0.....	14
3.2.4 Декомпозиция блоков диаграмм IDEF0.....	16
3.2.5 Отношения блоков на диаграммах модели IDEF0.....	19
3.2.6 Документирование модели IDEF0.....	21
3.3 Концептуальное проектирование учебной задачи.....	23
3.3.1 Контекстная диаграмма A-0 учебной задачи.....	23
3.3.2 Декомпозиция блока A0 учебной задачи.....	26
3.3.3 Декомпозиция процессов учебной задачи.....	28
3.3.4 Декомпозиция операций учебной задачи.....	30
3.3.5 Формирование ТТЗ учебной задачи.....	32
3.3.6 Техническое задание на ИС учебной задачи.....	33
3.3.7 Итог применения методологии IDEF0 к проектированию ИС.....	42
3.4 Моделирование потоков работ IDEF3.....	44
3.4.1 Элементы графического языка описания процессов IDEF3.....	44
3.4.2 Синтаксис и семантика языка IDEF3.....	48
3.4.2 Работы и перекрёстки работ методологии IDEF3.....	52
3.4.3 Документирование моделей IDEF3.....	53
3.4.4 Итоговая оценка применения методологии IDEF3.....	54
3.5 Моделирование потоков данных DFD.....	55
3.5.1 Синтаксис и семантика языка DFD.....	56
3.5.2 Постановка учебной задачи по методологии DFD.....	57
3.5.3 Формализация компонент контекстной диаграммы блока A23.....	58
3.5.4 Реализация контекстной диаграммы в системе Ramus Educational.....	59
3.5.5 Декомпозиция контекстной диаграммы учебной задачи.....	62
3.5.6 Итоговая одноуровневая диаграмма блока A23 в нотации DFD.....	63
3.6 Выводы по применению методологий концептуального проектирования ИС.....	65
Вопросы для самопроверки.....	66
ПЗ ЛАБОРАТОРНАЯ РАБОТА №3 Концептуальное проектирование ИС	
индивидуальной учебной задачи.....	67
ПЗ.1 Общая постановка задачи задачи.....	67
ПЗ.2 Содержательная часть лабораторной работы.....	67
ПЗ.3 Список использованных источников.....	67

3 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ ИС

Концептуальное проектирование ИС — это вторая стадия проектных работ, обеспечивающая перевод требований пользователей ИС на язык её разработчиков.

Как уже было отмечено ранее, в начале 1970-х годов стала пропагандироваться методика SADT и разрабатываться технологии интегрированного автоматизированного производства ICAM.

SADT (*Structured Analysis and Design Technique*) — методика структурного анализа и проектирования, пропагандируемая Дугласом Т. Россом.

ICAM (*Integrated Computer-Aided Manufacturing*) — программа разработки инструментов, методов и процессов для поддержки интеграции производства, запущенная в 1976 году департаментом ВВС США.

В конечном итоге проявленные усилия привели к созданию серии стандартов IDEF (ICAM Definitions), которые и определили основы различных методологий.

Что касается *тематики концептуального проектирования*, то сам этот термин имеет достаточно широкое толкование и применяется не только в технических системах. В частности, он используется при проектировании баз данных, что соответственно предполагает инфологическое проектирование их структуры и естественным образом затрагивает технологию построения семантической модели предметной области.

Основная задача данного раздела — изучение базовых методик, обеспечивающих процесс реализации второй стадии проектирования ИС.

Для успешной реализации поставленной задачи студенту необходимо:

- а) *перечитать* теоретическую часть подраздела 1.2, где описаны общепринятые стандарты, поддерживающие парадигму информационного подхода, а также раскрыты различия между функциональным и объектным аспектами описания предметной области;
- б) *повторить* содержимое пункта 1.3.3, в котором перечислены этапы работ по второй стадии проектирования АС;
- в) *руководствоваться* результатами проектных решений раздела 2, содержащих исходные данные для стадии концептуального проектирования.

Напомню, что согласно ГОСТ 34.601-90 [8], вторая стадия проектирования обозначается как «*Разработка концепции АС*», а в данном учебном пособии мы интерпретируем её как «*Концептуальное проектирование ИС*».

Результатом данной стадии должен быть отчёт, который в общем случае представляет собой «*Тактико-техническое Задание*» (**ТТЗ**) и на основе которого формируется, подписывается и утверждается документ «*Техническое задание*» (**ТЗ**).

Исходя из заявленной целевой установки, структура данной главы разделена на пять частей:

- а) подраздел 3.1 описывает стандартные модели, предназначенные для концептуального проектирования систем;
- б) подраздел 3.2 посвящён изучению функциональной модели стандарта IDEF0;
- в) в подразделе 3.3 рассматривается документальное оформление результатов второй и третьей стадий проектирования;
- г) в подразделе 3.4 изучается процессная модель стандарта IDEF3;
- д) в подразделе 3.5 представлен учебный материал по модели DFD.

3.1 Функциональное моделирование

Ещё раз отметим, что **в начале 1970-х годов** Дуглас Т. Росс стал пропагандировать методику «Структурный анализ и проектирование» (**SADT**, *Structured Analysis and Design Technique*), а **в конце 1970-х годов**, департамент военно-воздушных сил США реализовал идею SADT посредством стандартов **IDEF** (*ICAM Definitions*), где первой, если не главной, стала идея концепции функционального моделирования (**IDEF0**, *Function Modeling*).

Примечание — Важность концепции функционального моделирования состоит в том, что общие характеристики всех систем всегда оцениваются по набору функций, которые проектируемая система может исполнять.

Было бы удивительным рекомендовать для использования систему, которая не содержит некоторых важных функций. Перечень функций, пусть и неявный, содержится в любой постановке задачи. Более того, перечень функций системы обязательно перечисляется в техническом задании (ТЗ) и по этому перечню определяются результаты проектирования и реализации системы.

Примечание — Процесс проектирования любой системы — это процесс анализа её свойств от общего к частному (сверху-вниз).

Инструментарий стандарта IDEF0 идеально подходит для выполнения проектных работ на стадии концептуального проектирования ИС и включён в госстандарт России, как **ГОСТ Р 50.1.028-2001** [16].

Действительно:

- а) IDEF0 разработан на основе методики SADT и опирается на обобщённое понятие функции, преобразующей входные информационные или материальные объектные потоки в выходные потоки, учитывая при этом управляющие ограничения и «механизмы» необходимые для этого функционального преобразования;
- б) IDEF0 допускает декомпозицию функций до нужного уровня детализации, что позволяет удовлетворить основные требования специалистов, которые будут заниматься реализацией системы;
- в) IDEF0 позволяет проецировать иерархию функций модели на организационную структуру АС, что уже было показано ранее на рисунке 1.9 первого раздела (см. пункт 1.5.2).

Примечание — Инструментарий стандарта IDEF0 имеет и свои недостатки, которые устраняются другими моделями, такими как IDEF3 и DFD. В подразделах 3.4 и 3.5 они рассматриваются более подробно.

3.1.1 Функциональное моделирование бизнес-процессов

Функция — это не процесс. Функция обязана закончить свою работу и вернуть результат.

Понятие процесса (бизнес-процесса) является более сложным и комплексным, чем понятие функции. И прежде чем рассматривать моделирование функциями, кратко рассмотрим *методологии процессного моделирования*.

Процесс (бизнес-процесс) — это совокупность различных действий, работ или меро-

приятий человека или группы лиц, нацеленное на получение некоторого положительного результата для действующих акторов.

Привлекательность понятия бизнес-процесс для целей проектирования различных систем, включая ИС или АС, — вполне обоснована:

- а) *интуитивная понятность* для широкого круга специалистов и различных лиц, которым кажется, что — это понятно;
- б) *возможность различных уровней описательной абстракции*, позволяющей в будущем конкретизировать такие описания;
- в) *широкие возможности по манипулированию различными абстракциями*, скрывая за перманентностью самого процесса конкретизацию будущего результата.

В теоретическом плане выделяются *три целевых вида* процессных моделей:

- а) **Управляющие** — управляют функционированием систем, подобно таким как «Корпоративное управление» или «Стратегическое управление»;
- б) **Операционные** — составляют основной бизнес компании: «Снабжение», «Маркетинг» и другие подобные операционные действия;
- в) **Поддерживающие** — обслуживают основной бизнес-процесс: «Бухгалтерский учёт», «Техническая поддержка» и другие.

Примечание — **В частности**, во втором разделе учебного пособия, анализируя требования к бизнес-процессам в нашей учебной задаче (см. пункт 2.4.2), были выделены:

- а) **Занятие** — *главный бизнес-процесс*;
- б) **ВБП** — *второстепенные бизнес-процессы*, которые, в пункте 2.4.3, уточнены как: **ОГС** (операции с объектами *Группа-студентов*), **ОЭЖД** (операции с электронным журналом дисциплины), **ОПЗ** (операции подключения преподавателей к бизнес-процессу *Занятие*), **ОПС** (операции взаимодействия преподаватель-студент) и **ОСП** (операции взаимодействия студент-преподаватель).

Концепция процессного проектирования была подкреплена рядом модельных представлений, таких как:

- а) **BPM (Business Process Management)** — управление бизнес-процессами, которое поддерживается набором инструментальных средств, таким как **BPMS/BPMT (Business Process Management System/Tool)**;
- б) **BPMN (Business Process Model and Notation)** — нотация и модель бизнес-процессов, разработанная группой **Business Process Management Initiative**, поддерживается **OMG (Object Management Group)** и обеспечивает описание бизнес-моделей на языке XML;
- в) **BPEL (Business Process Execution Language)** — язык формального описания бизнес-процессов и протоколов их взаимодействия между собой на языке XML;
- г) **EPC (Event-driven Process Chain)** — диаграммы описания событийных цепочек бизнес-процессов, широко применяемых в системах планирования ресурсов предприятий (**ERP**);
- д) **IDEF3 (Integrated DEFinition for Process Description Capture Method)** — стандарт на методологию моделирования и документирования бизнес-процессов, происходящих в системе.

Особую популярность концепция процессного проектирования получила по результатам работ Августа-Вильгельма Шеера, разработавшего методологию моделирования бизнес-процессов предприятий и реализовавшего её в инструментальной системе ARIS (*Architecture of Integrated Information Systems, 1994 год*). Желаясь познакомиться с системой ARIS и другими подходами к проектированию рекомендуется прочесть обзорную статью [35].

Примечание — **Официальной методологией** структурного анализа и проектирования (*SADT*) является функциональный подход, описываемый стандартом *IDEF0*.

Понятие функции стандарта IDEF0 является более простым и конкретным конструктивом, чем понятие бизнес-процесса, описанного выше.

Функция — это преобразователь входных материальных или информационных объектов, предоставляющая на выходе достаточно конкретизированные объекты, которые могут быть использованы в дальнейшем для других функциональных преобразований. Само преобразование также должно иметь место. Если нет преобразования, то нет и функции.

Фактически, стандарт IDEF0 — это официальное описание методологии SADT, где функции являются основными элементами проекта, а бизнес-процессы описываются как различные комбинации выделенных функций.

Примечание — **Функциональный подход**, основанный на стандарте IDEF0, обеспечивает более чёткое проектное толкование бизнес-процессов, устраняя описанные выше негативные явления, присущие процессной методике проектирования.

В отличие от многих других стандартов, стандарт IDEF0 [16] имеет достаточно полное и подробное методологическое обеспечение, ориентированное на многоуровневое моделирование функций *таких систем как предприятие*. Методология IDEF0 даёт собственное более конструктивное толкование понятию «Бизнес-процесс», трактуя его как *конкретный уровень организационно-технической структуры предприятия* (см. учебный материал первого раздела: пункт 1.5.2 и рисунок 1.9):

- а) **Бизнес-процесс** — совокупность *Операций*;
- б) **Бизнес-процесс** — часть более общей производственной функции — *Деятельность*.

Рекомендации по стандартизации Р 50.1.028-2001 [16], разработанные комитетом ТК 431 «*CALS-технологии*», демонстрируют полный набор *структурных метамodelей*, которые показаны на рисунках 3.1 — 3.4.



Рисунок 3.1 — Метамодель деятельности предприятия [16]

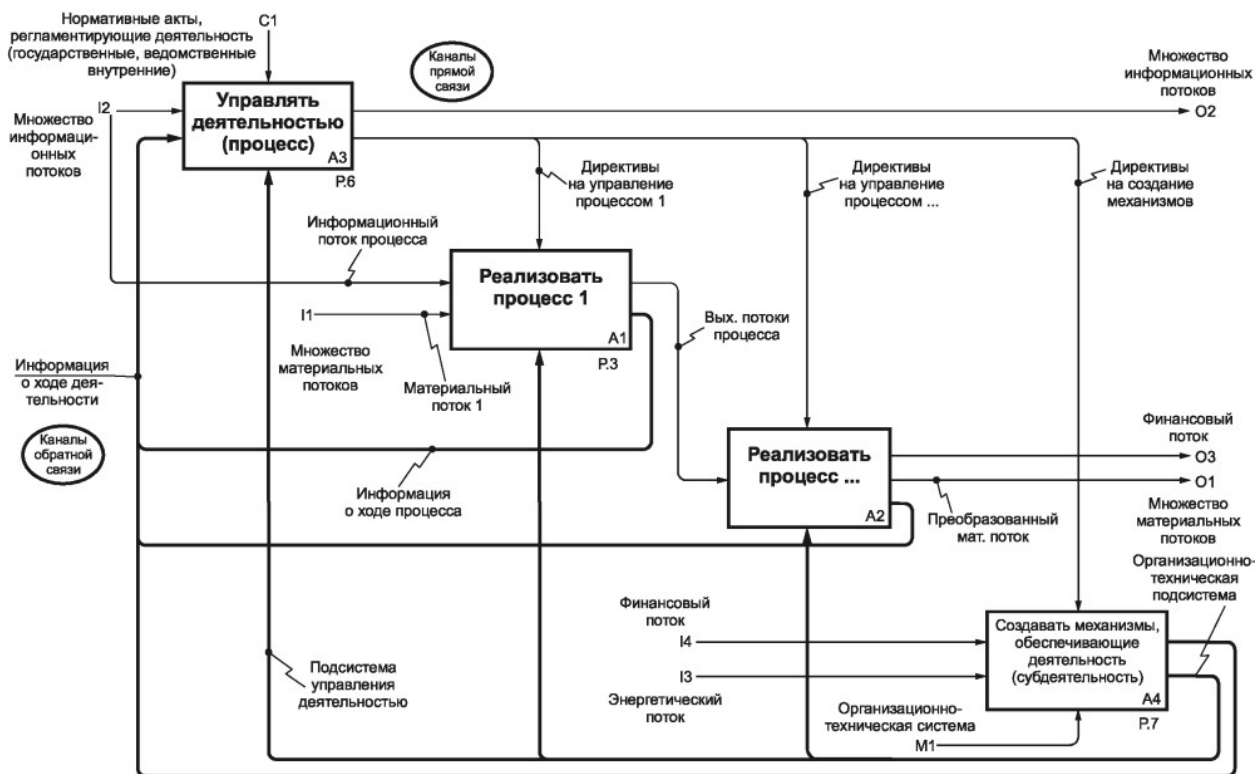


Рисунок 3.2 — Мета модель деятельности предприятия в виде процессов [16]

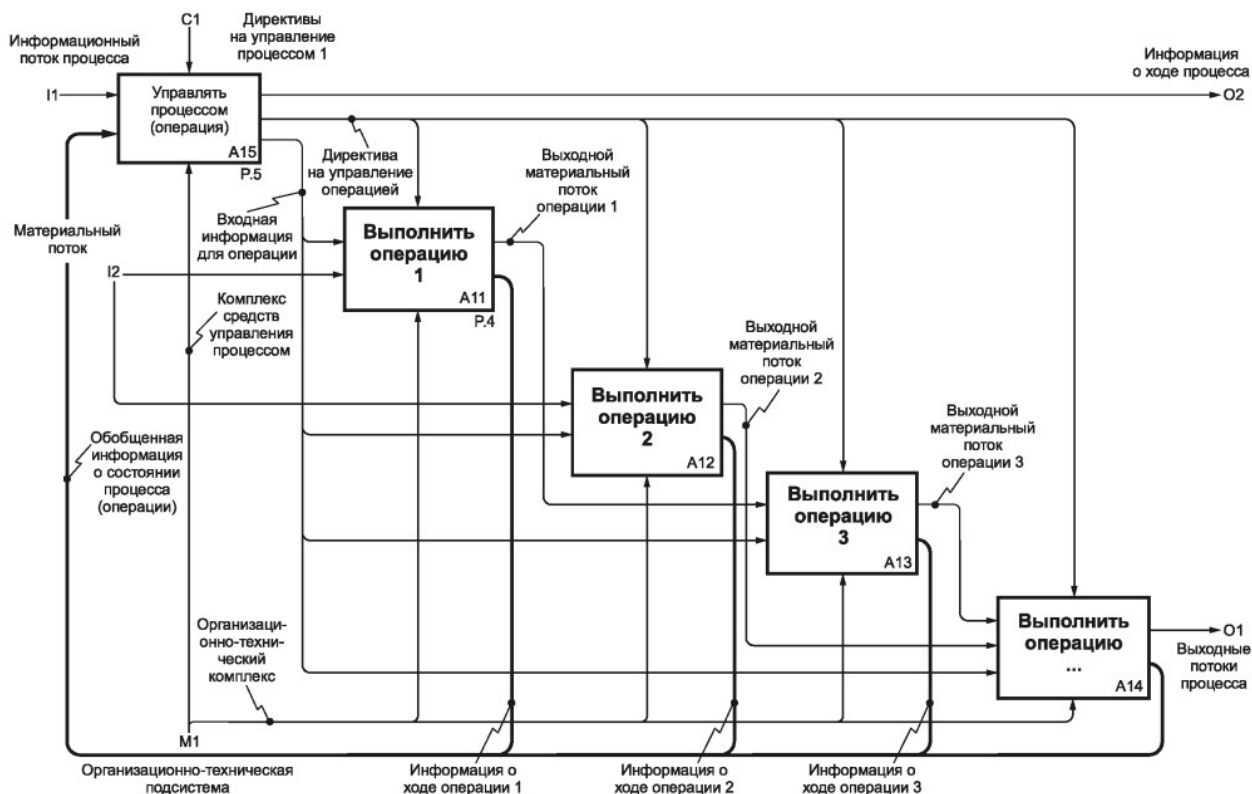


Рисунок 3.3 — Мета модель процесса предприятия в виде операций [16]

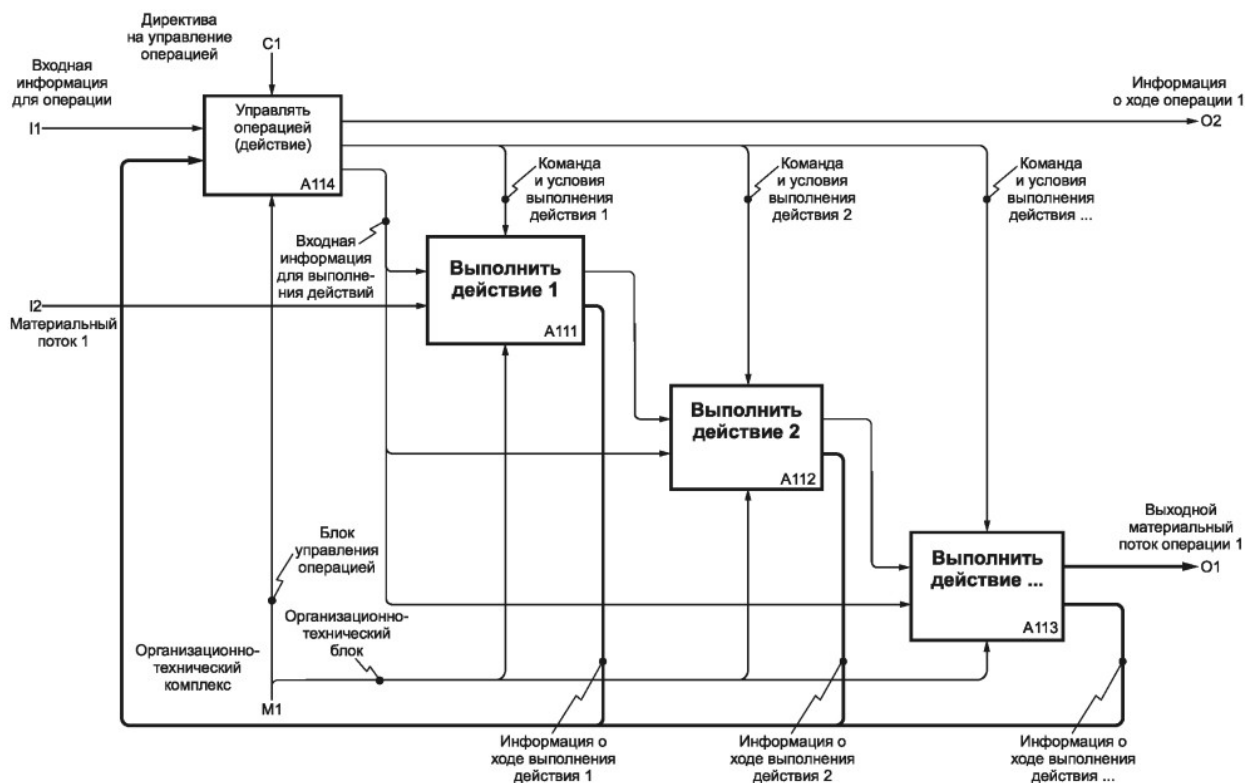


Рисунок 3.4 — Мета модель операции в виде действий [16]

Оставим подробную интерпретацию этих метамоделей до более подробного изучения синтаксиса и семантики методологии IDEF0, а сейчас обсудим место и значимость функциональных методологий на различных стадиях проектирования и создания ИС.

3.1.2 Применимость функциональных моделей к стадиям проектирования ИС

Общее понятие «Концептуальное проектирование» употребляется в двух семантических аспектах:

- 1) как *вторая стадия проектирования ИС (АС)*, определённая стандартом ГОСТ 34.601-90 [8];
- 2) как *абстрактное, но достаточно формализованное описание системы*, достаточное для последующего успешного её проектирования и реализации.

Первая интерпретация — вполне понятна. Она требует использования известных и понятных большинству экспертов формализованных моделей для написания ТТЗ (тактико-технического задания), достаточного для написания и последующего утверждения ТЗ (технического задания).

Вторая интерпретация требует ответа на вопрос: «*Какие модели и на каких стадиях, определённых ГОСТ 34.601-90 [8], должны быть использованы, чтобы успешно завершить создание системы ИС (АС)?*».

Хотя ответ на подобный вопрос всегда субъективен и содержит большую степень неопределённости, практика проектирования даёт следующий вполне приемлемый ответ.

Первая стадия создания ИС — «Формирование требований к АС» не предполагает обязательного представления каких-либо формализованных моделей, поэтому проектировщик берёт любые из них, по своему усмотрению.

Вторая стадия создания ИС ориентирована на конкурсное создание ТТЗ и последующее написание, согласование и утверждение ТЗ, поэтому основной моделью является струк-

турное моделирование по стандарту IDEF0.

Эскизный проект (четвёртая стадия) — использование полного набора имеющихся технологий моделирования, предполагающего анализ нескольких вариантов решений.

Технический проект (пятая стадия) предполагает возможную разработку ЧТЗ (частных технических заданий), что может потребовать использование моделей стандарта IDEF0.

Рабочая документация (шестая стадия) завершает реализацию всех программных частей ИС по уже созданным концептуальным решениям и не ориентирована на построение новых концепций.

Приведённые выше оценки применимости функциональных методологий сведены в представленную ниже таблицу 3.1.

Таблица 3.1 — Применимость функциональных моделей на разных стадиях создания ИС

<i>Стадия создания ИС</i>	<i>Применимость функциональных моделей</i>
1. Формирование требований к АС	Любые модели без ограничений.
2. Разработка концепции АС	IDEF0 — основная модель проектирования. IDEF3 — по мере необходимости. DFD — крайне редко.
3. Техническое задание	То же, что и на стадии 2
4. Эскизный проект	IDEF0 — основная модель проектирования. IDEF3 — основная модель проектирования. DFD — основная модель проектирования.
5. Технический проект	IDEF0 — по мере необходимости. IDEF3 — основная модель проектирования. DFD — основная модель проектирования.
6. Рабочая документация 7. Ввод в действие 8. Сопровождение АС	Не используются

Очевидно, что конкретная специфика проектируемых систем может изменить представленные оценки значимости тех или иных методологий. В таких случаях следует использовать более конкретные отраслевые стандарты и рекомендации.

Далее, учитывая особую роль стандарта IDEF0, изучим его применение в двух практических аспектах:

- как методологию функционального моделирования (проектирования) систем, главенствующую над методологией объектного проектирования;*
- как часть методологии концептуального проектирования ИС, на примере учебной задачи, требования к которой уже были описаны в предыдущем разделе данного пособия.*

3.2 Методология стандарта IDEF0

Базовая парадигма данного подраздела — модели стандарта IDEF0 являются основой методологии функционального подхода по проектированию ИС уровня предприятия.

В плане методологии, функциональный подход прежде всего противопоставляется объектному подходу:

- а) **функциональный подход** — методология концептуального проектирования систем;
- б) **объектный подход** — методология реализации систем.

Хотя функциональный подход и использует такие понятия как объекты и потоки объектов, в его основе лежит понятие функции, на что обращалось внимание как в первом разделе данного пособия, так и в учебном материале данного раздела.

Учебная цель данного подраздела — детальное изучение методологии стандарта IDEF0 в объёме, достаточном для самостоятельного применения студентами полученных знаний на всех стадиях проектирования ИС.

Базовым источником изучаемого здесь учебного материала являются Рекомендации по стандартизации Р 50.1.028-2001 [16], обязательные для изучения каждым студентом.

Формальным контекстом, в рамках которого осуществляется реальное проектирование ИС, является *стандартный состав проектной группы*.

3.2.1 Стандартный состав проектной группы

Реальная проектная работа связана с подготовкой множества различных документов и выполняется группой специалистов разной квалификации, что наглядно показано на рисунке 3.5.



Рисунок 3.5 — Рекомендуемый стандартный состав проектной группы [16]

Руководитель проекта является *административным лицом*, которое полностью отвечает за результат проектирования, состав и взаимодействие исполнителей.

Авторы (автор) — являются *основными исполнителями*, создающими модели IDEF0 на основе источников информации об объекте моделирования.

Библиотекарь проекта — *техническое лицо*, отвечающее за сохранность результатов проектирования и взаимодействие между *авторами, техническим советом и экспертами*.

Технический совет — *коллективный орган*, назначаемый руководителем проекта и обеспечивающий коллективные согласовательные решения как по всему проекту, так и по отдельным его частям.

Эксперты — лица, обладающие специальными знаниями об объекте моделирования и высказывающие полезные критические замечания.

Примечание — **Рекомендуемый состав проектной группы** легко масштабируется от её минимального необходимого состава до уровня проектной организации и сам этот состав может рассматриваться как объект автоматизации.

Таким образом, реальная проектная деятельность по методологии IDEF0 выполняется коллективом специалистов, в котором их функции и ответственность разделены в зависимости от умения и накопленного опыта.

В частности, руководитель практики от кафедры может конкретизировать схему рисунка 3.5 и отразить её в собственных методических рекомендациях по проведению производственной практики.

Организация учебного процесса студентов хоть и ориентирована на групповое обучение, но оценивается индивидуально. Поэтому студент выполняет и оформляет свой личный проект, обусловленный *конкретной проектной задачей*.

Технология создания IDEF0-моделей предполагает построение множества различных диаграмм и другой документации, которая явно функционирует в распределённой системе взаимодействующих субъектов. В этом отношении, автоматизация работы проектной группы может рассматриваться как задача на проектирование ИС.

3.2.2 Синтаксис и семантика языка IDEF0

Примечание — **Синтаксис языка IDEF0** определяется его компонентами — *блоки, стрелки, диаграммы и правила, дополненные связями между ними*.

Синтаксис языка IDEF0 полностью описан в пятом разделе документа [16].

Блок — *прямоугольный графический компонент* (см. рисунок 3.6), описывающий функцию *активным глаголом* или *глагольным оборотом*. В правом нижнем углу блока указывается его целочисленный идентифицирующий номер.

Стрелка — *графический именованный компонент*, обозначающий информационные или материальные объекты, состоящий из одного или последовательности непрерывных прямолинейных горизонтальных и вертикальных сегментов, сопряжённых дугами в 90 градусов, имеющий наконечник и соединяющий различные блоки. Допускаются стрелки присоединённые к блоку только одним концом.

Диаграмма — *графический компонент языка IDEF0*, размещённый на отдельном листе документа и содержащий один или несколько блоков, соединённых стрелками.

Правила — *определения*, указывающие как следует применять компоненты языка IDEF0.

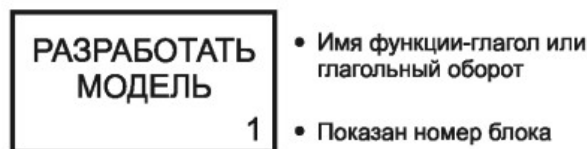


Рисунок 3.6 — Пример определения блока модели IDEF0 [16]

Примечание — Семантика языка IDEF0 определяет содержание (значение) и интерпретацию указанных выше синтаксических компонент языка.

Для раскрытия семантики языка в стандарте IDEF0 широко используется технология метаопределений, когда компоненты описываются словами, которые запрещены к употреблению в самих моделях. Покажем это конкретными примерами.

Блоки именуются *активными глагольными оборотами*, в которых не употребляется слово функция. Источник [16] прямо рекомендует примеры таких имён, представленных на рисунке 3.7.

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

Рисунок 3.7 — Примеры глагольных оборотов для именования блоков [16]

Стрелки, называемые также *дугами* или *интерфейсными дугами*, имеют гораздо большее разнообразие, что показано на рисунке 3.8.

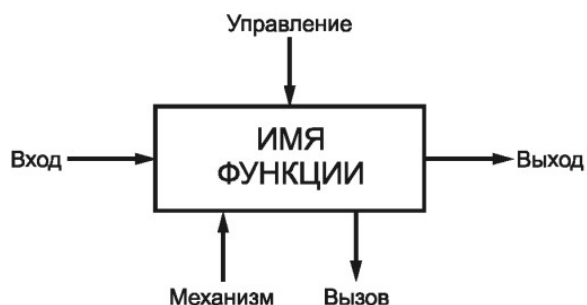


Рисунок 3.8 — Метаопределения стрелок модели IDEF0 [16]

Слова метаопределений стрелок также не должны присутствовать в их наименованиях. Источник [16] рекомендует использовать обороты, которые подобны примерам, показанным на рисунке 3.9.

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

Рисунок 3.9 — Примеры именования стрелок в моделях IDEF0 [16]

В целом, комбинированный пример отдельного блока показан на рисунке 3.10.

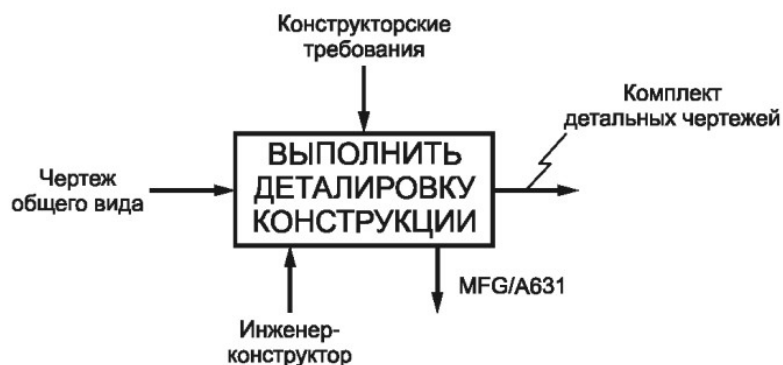


Рисунок 3.10 — Комбинированный пример блока со стрелками [16]

Правила — требования к именованию блоков и стрелок. Они дополнительно раскрывают смысловые значения *метаопределений стрелок*:

- а) **входные стрелки** должны касаться блоков с левой стороны;
- б) **выходные стрелки** должны выходить из блоков с правой стороны;
- в) **стрелки управления** должны подходить и касаться блоков в верхней их части;
- г) **стрелки механизмов** должны касаться блоков в нижней его части, причём *стрелка вызова* считается частью стрелки механизма;
- д) если имеются сомнения к какой стрелке относится метка её имени, то эта неопределённость устраняется молнеобразной линией, как показано на рисунке 3.10.

Диаграммы — главные компоненты модели IDEF0. Они содержат блоки, которые в общем случае соотносятся с функциями одного уровня: *деятельность, процесс, операция* или *действие* (см. функции представленные на рисунке 1.9 в пункте 1.5.2, а также диаграммы с метаопределениями, представленными на рисунках 3.1 - 3.4 в пункте 3.1.1).

Примечание — В целом, синтаксис и семантика модели IDEF0 — достаточно просты и ориентированы на однозначное понимание как разработчиками диаграмм, так и специалистами, которые читают и используют эти диаграммы.

К сожалению студенты, использующие модели IDEF0, допускают множество грубых ошибок, вызванных различными причинами. Автор данного пособия считает, что основная причина ошибок — *широкое использование учащимися объектных моделей*, которые при плохой подготовке студента не позволяют им достичь нужного понимания предмета. Поэтому дальнейшее *изложение учебного материала концентрируется на анализе ошибок*, часто допускаемых неопытным проектировщиком.

3.2.3 Контекстные диаграммы IDEF0

Каждая модель IDEF0 должна иметь контекстную диаграмму **A-0** (*A минус ноль*).

Практика показывает, что большинство студентов игнорируют построение контекстной диаграммы **A-0**, сразу представляя некоторую декомпозицию системы из нескольких блоков. *Это является грубой ошибкой*, потому что назначение контекстной диаграммы — *определить и зафиксировать границы моделируемой системы от внешней среды*, которая системой не является. В результате, если граница системы определена некачественно, то возникают различные проблемы и неопределённости с доказательной базой последующих проектных решений, а значит и дополнительные проблемы с её реализацией.

Примечание — В общем случае, стандартом IDEF0 допускается использование контекстных диаграмм *A-n*, где *n* — количество уровней иерархии модели, в пределах которой рассматривается проектируемая система, но на практике даже не обязательно указывать уровень *A-1*.

Обязательная контекстная диаграмма A-0 несёт следующую информационную нагрузку:

- а) **единственный блок диаграммы** определяет *границы моделируемой системы*, концентрируя в названии блока всю функциональность её модели;
- б) **стрелки этой диаграммы** не принадлежат моделируемой системе, а *отображают связи модели с внешним миром*, полностью ограничивая и её *внешний интерфейс*;
- в) **ЦЕЛЬ:** — причина создания модели IDEF0, содержащая набор важных свойств, которые эта модель должна отображать;
- г) **ТОЧКА ЗРЕНИЯ:** — должностное лицо или группа лиц, для которых создаётся или создана модель IDEF0.

Для демонстрации контекстной диаграммы *A-0* используем пример из источника [16], который показан на рисунке 3.11 и исполнен в инструментальной системе Ramus Educational [27].

Примечание — **Грубой ошибкой** считается *отсутствие на диаграмме A-0 «ЦЕЛИ:»* или повторение в ней названия функционального блока A0.

Следует помнить, что любая модель, включая IDEF0, только приближённо описывает реально существующую систему, поэтому *«ЦЕЛЬ:» задаёт ту нужную проекцию свойств системы*, которые будут наиболее важными на диаграммах и в пределах которых проводить последующую декомпозицию контекстной диаграммы. Например, на рисунке 3.11 явно указывается, что последующая декомпозиция A0 будет содержать функции менеджера, возглавляющего *«Бригаду программистов»*.

Также должны быть отражены все информационные потоки, присутствующие в процессах создания *«Программы»*, по которым можно будет оценивать трудоёмкость её создания. Обычно, более подробная информация о *«ЦЕЛИ:»* раскрывается в текстовом документе, поясняющем назначение компонентов модели.

Что касается нашей учебной задачи, рассмотренной в следующем подразделе, то её *«ЦЕЛЬ:» — обеспечить стадию концептуального (структурного) проектирования ИС*.

Примечание — **Грубой ошибкой** считается отсутствие на диаграмме A-0 *«ТОЧКИ ЗРЕНИЯ:»* или указание в качестве неё позиции автора (проектировщика).

Обычно студент указывает, что это — его точка зрения. Необходимо помнить, что автор (проектировщик) является *исполнителем*, а не *заказчиком модели*.

Точку зрения формулирует заказчик модели, что требует от автора учёта всех его профессиональных особенностей и интересов в получении результата моделирования. В таких условиях необходимо обратиться к организационной структуре предприятия, исследованной на первой стадии проектирования, и указать соответствующее должностное лицо или группу лиц, которые будут оценивать проект.

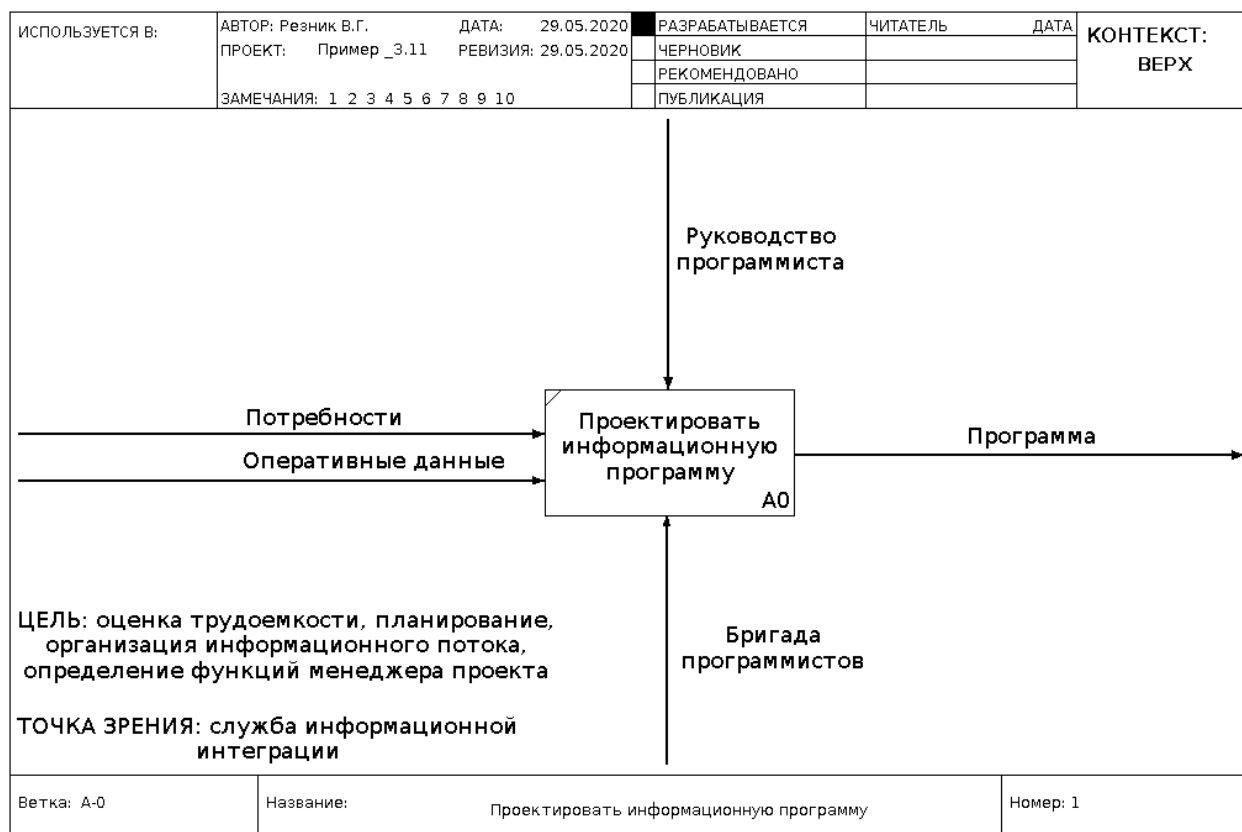


Рисунок 3.11 — Пример контекстной диаграммы [16]

3.2.4 Декомпозиция блоков диаграмм IDEF0

Обязательный состав модели IDEF0 — три типа документов: *набор графических диаграмм, глоссарий и текст.*

Набор графических диаграмм — основной результат структурного представления модели IDEF0, связанных родительскими и дочерними отношениями.

Глоссарий — документ, содержащий в алфавитном порядке список сокращений или наборов слов, использованных в именах блоков и стрелок, а также раскрывающий их точное смысловое содержание.

Текст — документ, раскрывающий и комментирующий в текстовом виде графические изображения диаграмм.

Декомпозиция блоков диаграмм тесно связана с родительскими и дочерними соотношениями функций модели IDEF0.

Родительская диаграмма — диаграмма, содержащая блок, который подвергнут декомпозиции на текущей диаграмме.

Дочерняя диаграмма — диаграмма, которая содержит декомпозицию рассматриваемого функционального блока.

Общая архитектурная декомпозиция блоков диаграмм модели IDEF0 показана на рисунке 3.12, заимствованном из источника [16].

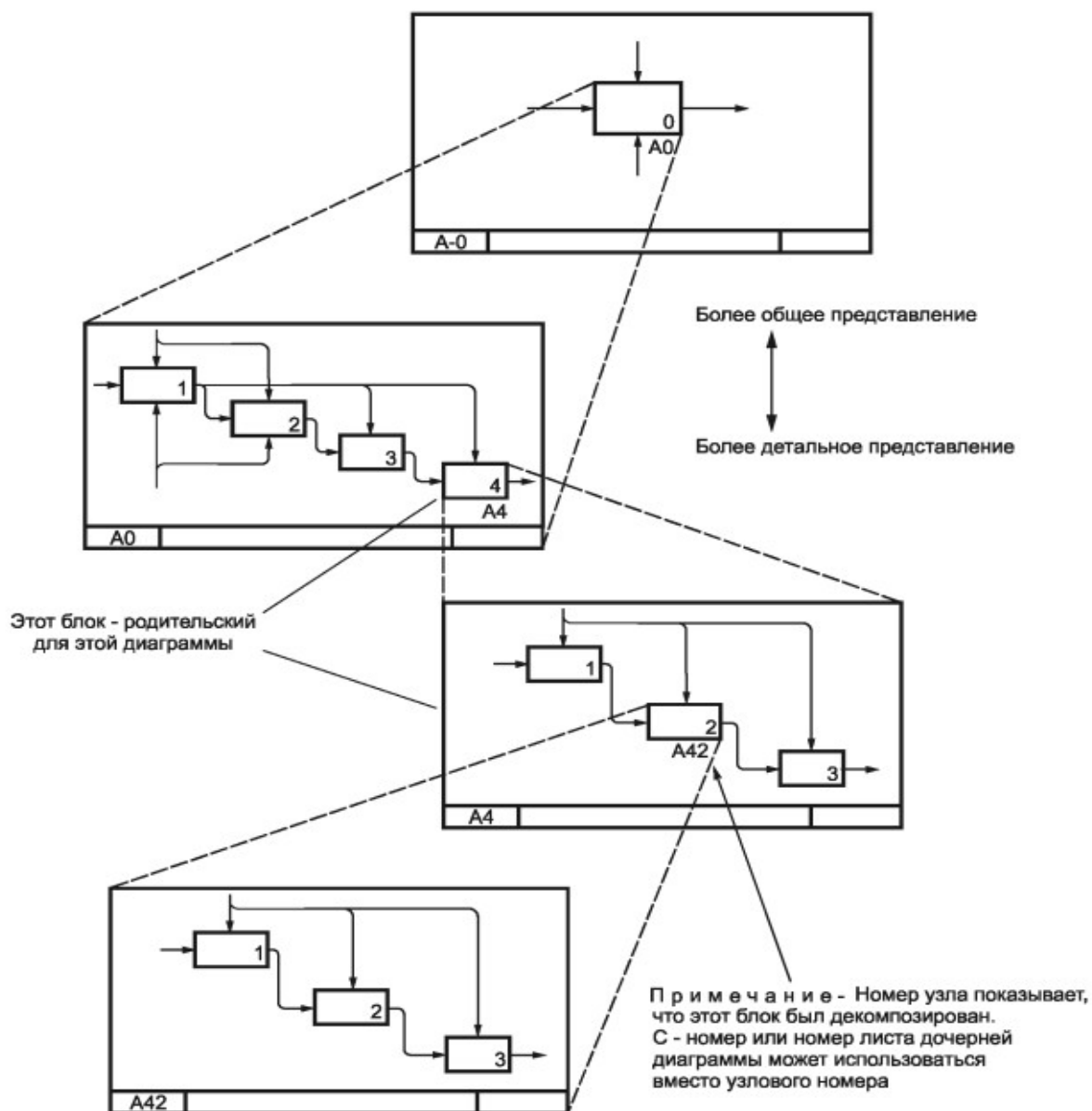


Рисунок 3.12 — Общая архитектурная декомпозиция блоков диаграмм [16]

Примечание — Все наименования блоков в наборе графических диаграмм должны быть различными, а наименования стрелок блоков родительских диаграмм передаются в дочерние диаграммы.

Первой выполняется декомпозиция единственного блока A0 контекстной диаграммы A-0. При этом, каждый отдельный блок диаграммы A0 должен иметь меньшую функциональность, чем сам блок диаграммы A-0, а в сумме они должны точно соответствовать его функциональности. Желательно, чтобы все блоки диаграммы A0 имели один уровень функциональности и имели чёткие проекции на классификацию: *деятельность, субдеятельность, процесс, подпроцесс, операция и действие*.

Таким образом, декомпозиция блоков обеспечивает построение иерархической модели целевой системы с нужным уровнем детализации, который определяется *не проектировщиком, а заказчиком моделируемой системы*.

Применительно к тематике нашей дисциплины, модель IDEF0 должна обеспечить стадию концептуального проектирования целевой ИС, включая её отчётную часть. И чтобы реа-

лизовать *целостность* и *полезную конструктивность* такой модели, используются средства точной *адресации* блоков диаграмм, *ICOM-кодирование* и *туннелирование* их дуг.

Адресация блоков диаграмм — сквозное средство идентификации компонент модели IDEF0, осуществляемое в двух аспектах:

- а) **внутреннее кодирование блоков** определяется в пределах самой диаграммы, начиная с 1 и заканчивая их общим числом; *рекомендуется использовать не более шести блоков*; исключением является блок диаграммы A-0, который кодируется номером 0;
- б) **внешнее кодирование блоков** определяется в пределах всей модели, как это показано на рисунке 3.13.

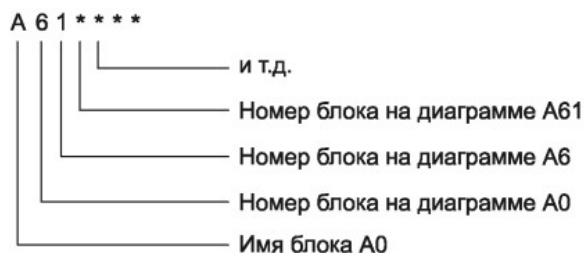


Рисунок 3.13 — Правила кодирования блоков и диаграмм модели IDEF0 [16]

ICOM-кодирование дуг — *дополнительное кодирование имён стрелок* (меток стрелок), связывающее граничные стрелки на дочерней диаграмме со стрелками родительского блока.

Синтаксис ICOM-кода представляет собой одно из буквенных обозначений I, C, O или M, после которого указывается цифра в диапазоне от 1 до максимального количества стрелок на соответствующей стороне родительского блока, увеличивающаяся по правилу: *сверху-вниз и слева-направо*.

Семантика используемых буквенных обозначений:

- а) **In** — *n-я сверху стрелка входа (Input)*;
- б) **Cn** — *n-я слева стрелка управления (Control)*;
- в) **On** — *n-я сверху стрелка выхода (Output)*;
- г) **Mn** — *n-я слева стрелка механизма (Mechanism)*.

Туннелирование дуг — дополнительный графический элемент для начала или конца стрелки в виде круглых скобок, имеющий собственную семантику.

Стрелка, обозначающая туннель в месте подключения к блоку, — представляет объекты, *не присутствующие* на следующем уровне декомпозиции (см. рисунок 3.14).

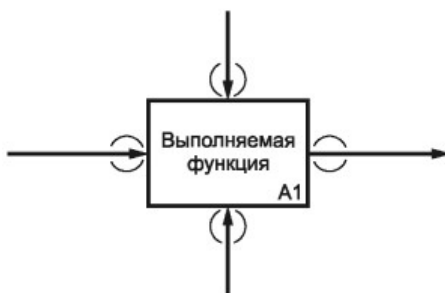


Рисунок 3.14 — Стрелки не присутствующие в декомпозиции блока [16]

Стрелка, обозначающая туннель на своём свободном конце, — обозначает представляемые ею объекты, *отсутствующие* на родительской диаграмме (см. рисунок 3.15).

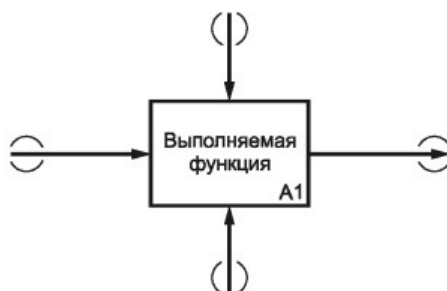


Рисунок 3.15 — Стрелки не присутствующие на родительской диаграмме [16]

Необходимость туннелирования стрелок вызвана необходимостью удаления из диаграммы «лишних» компонент, затрудняющих её восприятие.

3.2.5 Отношения блоков на диаграммах модели IDEF0

Семантическая выразительность графического языка IDEF0 определяется *шестью типами отношений* между блоками каждой отдельной диаграммы: *доминирование, управление, выход-вход, обратные связи по входу и выходу, выход-механизм*.

Доминирование — *отношение важности (доминирования) блоков своим расположением на диаграмме*. Более важными являются блоки, расположенные выше и левее, например, как это показано на рисунке 3.12 (см. пункт 3.2.4). Это также соответствует первоначальному расположению блоков на диаграмме: «лесенкой» сверху-вниз и слева-направо. В таком же порядке и нумеруются блоки на диаграмме.

Управление — *отношение прямого управляющего воздействия*, когда выход какого-либо блока является управляющим входом на блок с меньшим доминированием (см. рисунок 3.16).

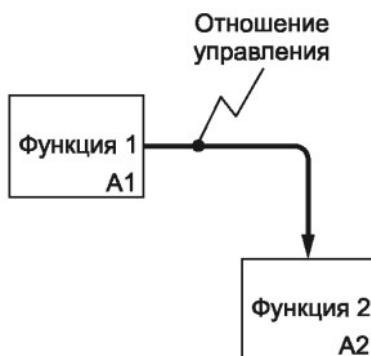


Рисунок 3.16 — Отношение управления [16]

Выход-вход — *отношение прямой передачи объектов между блоками*, когда выход какого-либо блока поступает на вход другого блока с меньшим доминированием (см. рисунок 3.17).

Обратная связь по управлению — *отношение, задающее циклическую обратную связь*, когда выход одного блока поступает на управляющий вход другого блока с большим доминированием (см. рисунок 3.18).

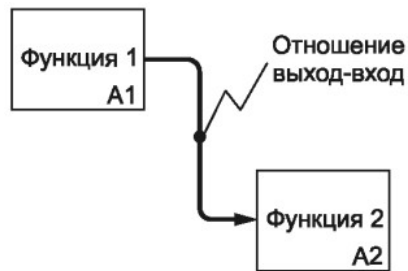


Рисунок 3.17 — Отношение выход-вход [16]



Рисунок 3.18 — Отношение обратной связи по управлению [16]

Обратная связь по входу — отношение, задающее циклическую обратную связь, когда выход какого-либо блока поступает на вход другого блока с большим доминированием (см. рисунок 3.19).

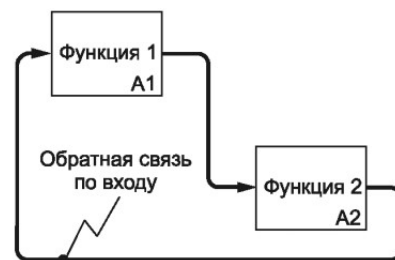


Рисунок 3.19 — Отношение обратной связи по входу [16]

Выход-механизм — наиболее сложное отношение, задающее каким-либо блоком средства для реализации функции другого блока с меньшим доминированием (см. рисунок 3.20).

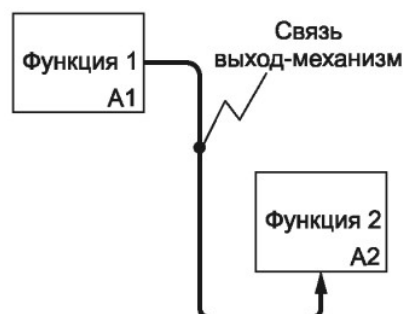


Рисунок 3.20 — Отношение выход-механизм [16]

Примечание — Допускается наличие циклических обратных связей для одного блока, как это показано на рисунке 3.21.

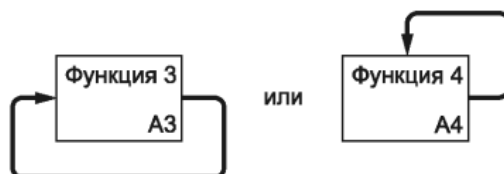


Рисунок 3.21 — Примеры двух видов циклических обратных связей для одного блока [16]

Примечание — В общем случае, между блоками допускаются комбинированные связи, включающие различные типы отношений.

Проблема наличия множества отношений между блоками является всегда актуальной, поэтому *рекомендуется на диаграмме размещать не более шести функциональных блоков*.

3.2.6 Документирование модели IDEF0

Официально, модель IDEF0 требует наличия трёх типов документов: *набора графических диаграмм, глоссария и текста*. Этот набор документов может оформляться как самостоятельно, так и в составе других документальных пакетов, например, в составе ТЗ на разработку ИС. Поскольку восприятие большого количества страниц диаграмм является достаточно затруднительным занятием, источник [16] рекомендует использовать такие дополнительные представления, как *перечень узлов и дерево узлов*.

Перечень узлов — *дополнительное средство документирования* части модели IDEF0, представленное в виде форматированного списка. Пример такого представления для некоторой абстрактной модели показан на рисунке 3.22.

- A0 Производить продукт
 - 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 Разрабатывать 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

Рисунок 3.22 — Пример перечня узлов [16]

Обратите внимание, что *перечень узлов* легко может быть представлен в электронном виде, например, в формате файла XML, в формате файла JSON или перечня пунктов меню некоторого интерфейса пользовательского приложения.

Дерево узлов — *дополнительное средство документирования* части модели IDEF0, представленное в графическом виде, например, как это показано для той же абстрактной модели на рисунке 3.23.

Соответствующее электронное представление дерева узлов также может быть использовано в интерфейсах пользовательских приложений.



Рисунок 3.23 — Пример дерева узлов [16]

К дополнительным средствам документирования следует отнести и диаграммы *FEO*.

FEO (For Exposition Only) — *диаграммы-иллюстрации*, которые основаны на стандартных диаграммах методологии IDEF0, но имеют различные поясняющие «украшательства», например, различный цвет блоков и стрелок, а также дополнительные обозначения и поясняющие элементы.

В целом, диаграммы FEO не являются обязательной частью документации методологии IDEF0, а используются для публичных презентаций или иных демонстрационных мероприятий.

Примечание — Все диаграммы стандарта методологии IDEF0 должны быть исполнены в чёрно-белом исполнении и не допускать посторонних форматирований и дополнительных обозначений, кроме упомянутых выше.

Завершая данный подраздел, особо отметим, что отображение стрелок на диаграммах должно строго удовлетворять следующим **общим правилам**:

- а) *все отдельные сегменты стрелок* должны иметь только горизонтальное или вертикальное начертание и соединяться с *чёткими закруглениями*;
- б) *пересечения различных стрелок* должны осуществляться только под прямым углом и не допускают каких-либо дополнительных начертаний;
- в) *допускается ветвление стрелок*, демонстрируя параллельное распространение материальных или информационных объектов;
- г) *допускается слияние стрелок*, демонстрируя объединение различных материальных и информационных объектов в один поток.

Таким образом, в данном подразделе изложены все основные синтаксические и семантические аспекты теоретической части методологии IDEF0. Все дополнительные нюансы конкретного применения этой теории, например, ICOM-кодирование граничных стрелок следует уточнять по источнику [16].

Теперь перейдём к практической демонстрации использования методологии IDEF0 для целей концептуального проектирования ИС.

3.3 Концептуальное проектирование учебной задачи

В предыдущем подразделе были кратко описаны теоретические основы построения моделей по стандарту IDEF0.

Цель данного подраздела — показать как теория структурного анализа и проектирования (SADT), представленная стандартом IDEF0, может быть применена для целей концептуального проектирования ИС (вторая стадия проектирования АС).

Целевая задача на проектирование была сформирована во втором разделе данного пособия (см. подраздел 2.1), где также была выполнена *первая стадия* проектирования информационной системы «*Электронный журнал руководителя*» (сокращено — ЭЖР).

Цель второй стадии проектирования — построение концептуальной модели ИС ЭЖР, используя формализацию поставленной задачи и исходных материалов собранных на первой стадии проектирования, достаточной для отчётности в виде ТТЗ и последующей реализации третьей стадии проектирования — написание и утверждение ТЗ на ИС ЭЖР.

Исполнителем второй стадии проектирования назначен абстрактный студент группы 447-1 Петров И.В., *поэтому все претензии по недостаткам проекта — к нему!*

Руководителем второй стадии проектирования является *автор данного пособия*, задача которого исправлять и комментировать ошибки, допущенные **Исполнителем**.

Примечание — **Методология концептуального проектирования**, выполняемая по стандарту IDEF0, требует для проектируемой ИС строгой формализации объектов и функций высшего уровня, а также — представления этой формализации в виде контекстной диаграммы А-0, с указанием «ЦЕЛИ:» и «ТОЧКИ ЗРЕНИЯ:» всего проекта.

Для лучшего разделения всей последовательности работ по проектированию, изложим учебный материал в отдельных пунктах, выделив:

- а) построение контекстной диаграммы (см. пункт 3.3.1);
- б) несколько этапов декомпозиции (см. пункты 3.3.2-3.3.4);
- в) формирование ТТЗ учебной задачи (см. пункт 3.3.5);
- г) формирование ТЗ учебной задачи (см. пункт 3.3.6).

В качестве основного инструментального средства для построения всех диаграмм проекта будет использоваться система Ramus Educational.

3.3.1 Контекстная диаграмма А-0 учебной задачи

Общие требования к контекстной диаграмме А-0 и демонстрирующий её пример подробно изложены в предыдущем подразделе (см. пункт 3.2.3).

Построение контекстной диаграммы А-0 для учебной задачи требуют определения *границ* функционального блока А0, *имени* этого блока и *организационного уровня* его функции, что необходимо зафиксировать в текстовом документе, являющимся частью модели IDEF0.

Имя проекта (в инструментальной системе Ramus) зафиксируем как «**ИС ЭЖР**».

Организационный уровень блока А0 определим как *субдеятельность*, поскольку обучение студентов является частью деятельности всего вуза.

Имя блока А0 зафиксируем как «**Выполнить семестровую учебную нагрузку препода-**

давателя», поскольку все занятия преподаватель выполняет в соответствии с заранее выделенным ему *планом*, включающим: *перечень дисциплин и учебных групп*, закреплённых за ним, а также — *расписание по всем видам занятий*, которые включены в эти дисциплины.

Сказанное выше уже определяет *внешний ресурс реализации блока А0* и *внешние управляющие ограничения* этой реализации:

- а) *стрелка механизма* — *внешний ресурс* в виде объекта «*Преподаватель*»;
- б) *стрелки управления* — *внешние ограничения* в виде информационных объектов: «*Учебная нагрузка*», «*Расписание занятий*» и «*УМП дисциплин*».

Внешние объекты на входе — это *внешние физические объекты* — «*Студенты на обучение*» и *внешние информационные объекты* — «*Данные о студентах*».

Внешние объекты (стрелки) на выходе — *физические объекты* «*Обученные студенты*» и «*Студенты без доступа*», плюс *внешние информационные объекты* — «*Оценки обучения студентов*».

Окончательно, дополняем диаграмму А-0 *целью и точкой зрения*:

- а) **ЦЕЛЬ:** концептуальное проектирование системы ИС ЭЖР;
- б) **ТОЧКА ЗРЕНИЯ:** заведующий кафедрой АСУ ТУСУР.

На основе проведённой формализации, Петров И.В. построил контекстную диаграмму А-0 в системе Ramus, которая показана на рисунке 3.24.

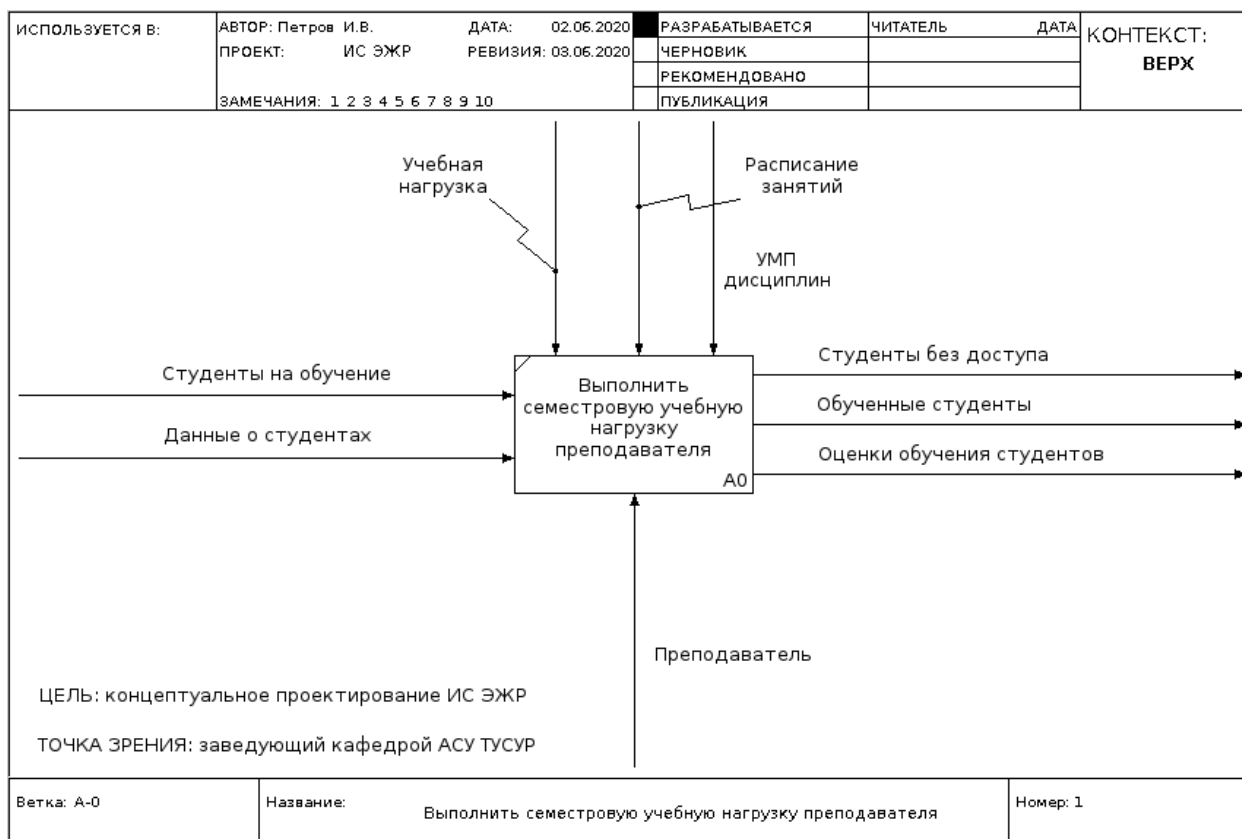


Рисунок 3.24 — Контекстная диаграмма А-0 учебной задачи

Студент Петров И.В. максимально точно и правильно построил контекстную диаграмму А-0, но на текущий момент эта диаграмма вполне понятна только самому студенту, его руководителю и возможно многим преподавателям, имеющим достаточный опыт проектирования информационных систем.

Примечание — **Высокий уровень абстракции** семантических значений компонент контекстной диаграммы А-0 порождает большую *неопределённость в толковании понятий концептуального проектирования*.

Свойства человеческого мышления — таковы, что каждый из читателей представленной диаграммы А-0 понимает её по своему, внося своё видение как в систему в целом, так и в последующую её детальную декомпозицию. В последующем, такое различное видение свойств системы и её деталей приведёт уже к явным противоречиям, которые будут активно высказываться всеми, кто читает и анализирует диаграммы проекта.

Чтобы максимально избежать последствий описанной выше общей проблемы, Петрову И.В. поручается: дополнить текстовое описание контекстной диаграммы А-0 подробным анализом того факта, что проектируемая система является *индивидуальным средством автоматизации субъективности преподавателя (руководителя)*, поэтому она запускается им только периодически и для достижения следующих трёх целей:

- 1) *реализации подготовительной части работ*, связанной с формированием списков групп студентов на основе входного потока (стрелки) «**Данные о студентах**» и для связи этих групп с дисциплинами, входящими во внешний ограничительный документ (стрелку) «**Учебная нагрузка**»;
- 2) *реализации заключительной части работ*, связанной с формированием внешнего выходного потока (стрелки) «**Оценки обучения студентов**»;
- 3) *реализации базовой части работ*, осуществляемой согласно заданному внешнему ограничению (стрелке) «**Расписание занятий**» для преобразования на каждом отдельном занятии внешнего входного потока «**Студенты на обучение**» во внешний выходной поток «**Обученные студенты**».

Примечание — **Основным документом**, обеспечивающим единое семантическое толкование компонент всех диаграмм проекта, является правильно составленный *Глоссарий*.

Текстовое описание диаграммы проекта может содержать достаточное и подробное описание смыслового содержания диаграммы, но воспринимать этот текст сложно, особенно, если речь идёт о спорном толковании его отдельных компонент. Указанный недостаток семантики проекта должен устранять *глоссарий*, описывающий каждую компоненту диаграммы настолько подробно, чтобы не допустить неоднозначного толкования этого описания.

Полный глоссарий по всем диаграммам должен быть оформлен отдельным документом, содержащим список всех использованных наименований в алфавитном порядке, чтобы было удобно их находить. В данном же пункте излагаются только понятия, относящиеся к приведённой выше контекстной диаграмме А-0. *Например*.

Выполнить семестровую учебную нагрузку преподавателя — *субъективность преподавателя*, именуемая выполняемой функцией ИС ЭЖР.

Сама субъективность осуществляется посредством периодического запуска ИС:

- а) как строго по расписанию занятий для выполнения базовой части работ;
- б) так и в произвольное время для выполнения подготовительной и заключительной части работ.

Данные о студентах — *внешние информационные объекты*, уникально идентифицирующие каждого студента, включая принадлежность к группе обучения.

Обученные студенты — *внешние физические объекты на выходе ИС ЭЖР*, которые циклически подвергаются субдеятельности преподавателя и получают на отдельном занятии дисциплины порцию знаний, предусмотренную внешними управляющими ограничениями субдеятельности.

Оценки обучения студентов — *внешние информационные объекты на выходе ИС ЭЖР*, соответствующие результатам аттестации преподавателем знаний студентов.

Преподаватель — *внешний механизм ИС ЭЖР*, с помощью которого осуществляется функция субдеятельности проектируемой системы.

Расписание занятий — *внешнее управляющее воздействие*, определяющее дату, время и продолжительность отдельной порции субдеятельности преподавателя.

Студенты без доступа — *выходные физические объекты ИС ЭЖР*, соответствующие входным объектам «*Студенты на обучение*», которые по каким-либо причинам не получили доступ к занятию (конкретной порции субдеятельности преподавателя) и не получили (предусмотренную занятием) порцию знаний.

Студенты на обучение — *входные физические объекты ИС ЭЖР*, которые преобразуются блоком А0 диаграммы А-0 в два внешних физических потока объектов: «*Студенты без доступа*» и «*Обученные студенты*».

УМП дисциплин — *внешние управляющие объекты*, определяющие объём знаний для каждой порции субдеятельности преподавателя.

Учебная нагрузка — *внешние управляющие объекты блока А-0*, ограничивающие состав и объём субдеятельности преподавателя.

Примечание — Методика IDEF0 требует полного завершения процесса создания контекстной диаграммы А-0 перед началом работ по её декомпозиции.

3.3.2 Декомпозиция блока А0 учебной задачи

Первая задача декомпозиции любого блока модели IDEF0 — определение числа входящих в него блоков и их функциональное назначение.

Любая инструментальная система, при попытке выполнить декомпозицию выделенного блока *спросит о количестве блоков, входящих в декомпозицию*, а затем создаст новую страницу для диаграммы и разместит в ней неименованные блоки *в порядке их доминирования*. Поэтому общая практика создания диаграмм модели IDEF0 состоит в первоначальном использовании инструментов в виде *бумаги, карандаша, линейки и стирательной резинки*.

Что касается декомпозиции блока А0 учебной задачи, то его декомпозиция по блокам уже фактически была заявлена в предыдущем пункте как *совокупность трёх процессов*:

- а) *подготовительная часть работ*;
- б) *базовая часть работ*;
- в) *заключительная часть работ*.

Примечание — **Функциональное название любого блока диаграммы** должно указывать, как этот блок «перерабатывает» каждую отдельную совокупность объектов входного потока в другую совокупность объектов выходного потока.

Дополним уже начатый документ глоссария тремя названиями блоков, входящих в диаграммы А0:

- 1) *подготовить список группы и ЭЖД* (Электронный Журнал Дисциплины);

- 2) *осуществить проведение занятия;*
- 3) *оценить работу студента.*

Подготовить список группы и ЭЖД (A1) — функция *процесса преподавателя*, входящая в его субъектность и соответствующая подготовительной части его работ, которая осуществляется по управляющему ограничению «**Учебная нагрузка**» и преобразует входной информационный поток объектов «**Данные о студентах**», сначала в объекты списков групп, а затем, — в объекты «**ЭЖД занятий**». ЭЖД — *Электронный журнал дисциплины*.

Осуществить проведение занятия (A2) — функция *процесса преподавателя*, входящая в его субъектность и соответствующая базовой части его работ, которая осуществляется под управляющими ограничениями «**Расписание занятий**» и «**УМП дисциплин**». Эта функция преобразует входные потоки «**ЭЖД занятий**» и «**Студенты на обучение**» в четыре выходных потока: «**Студенты без доступа**», «**Обученные студенты**», «**Отчёты студентов**» и «**ЭЖД после занятия**».

Оценить работу студента (A3) — функция *процесса преподавателя*, входящая в его субъектность и соответствующая заключительной части его работ, которая выполняется под управляющим ограничением «**УМП дисциплин**». Она преобразует входные потоки «**ЭЖД после занятия**» и «**Отчёты студентов**» в два выходных потока: «**Оценки обучения студентов**» и «**ЭЖД после оценки отчётов**».

На основании выделенных блоков, диаграмма A0 — построена и показана на рисунке 3.25, а глоссарий дополняется определениями внутренних стрелок этой диаграммы.

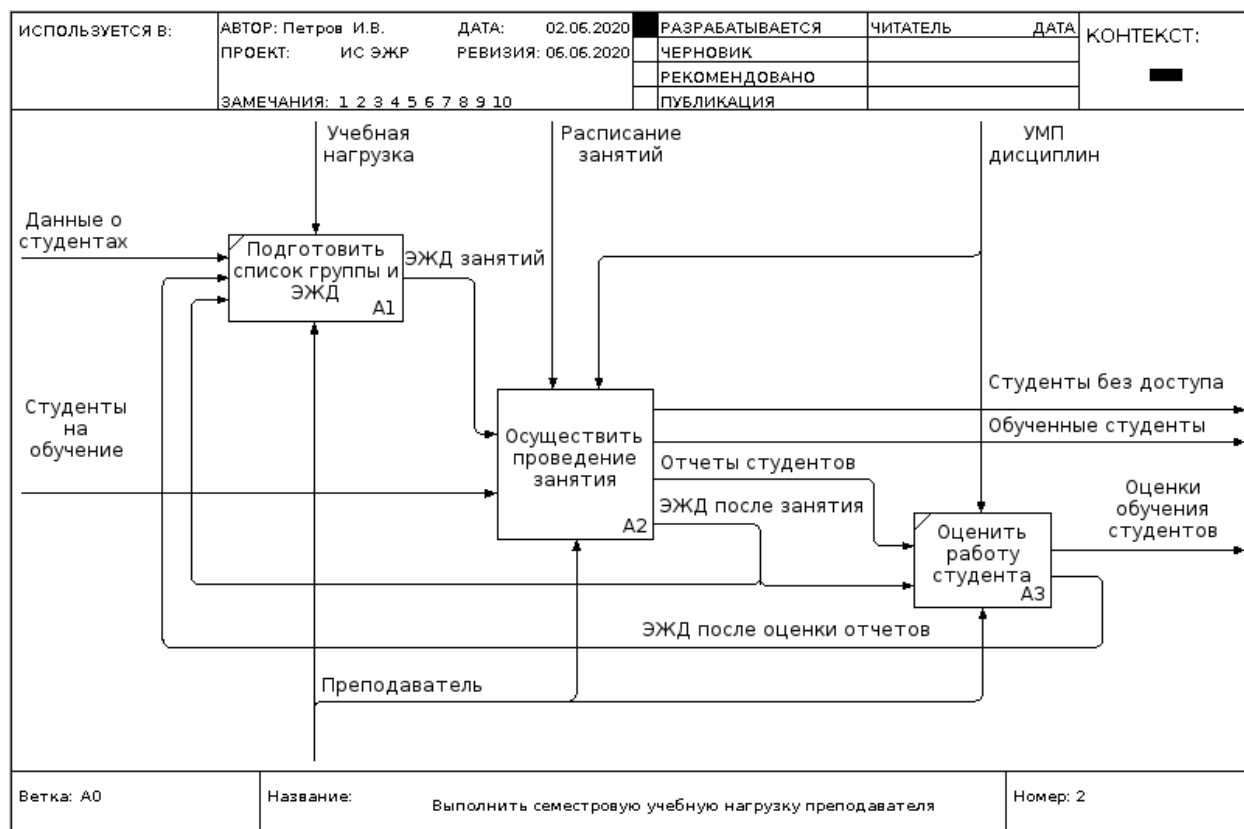


Рисунок 3.25 — Диаграмма A0, отображающая процессы ИС ЭЖР

Отчёты студентов — *индивидуальные документы студентов*, оформляемые ими в виде файлов или сообщений по результатам проведённого занятия и доступные (в каком-либо виде) преподавателю для оценивания.

ЭЖД занятий — *внутренние данные ИС ЭЖР*, формируемые для отдельных дисциплин.

плин учебной нагрузки на основе уже сформированных списков групп. Отдельный (для каждой дисциплины) ЭЖД хранится в отдельной базе данных дисциплины.

ЭЖД после занятия — *отдельный (для каждого занятия) ЭЖД*, в который добавлена информация о проведённом занятии.

ЭЖД после оценки отчётов — *отдельный (для каждого занятия) ЭЖД*, в который добавлена информация об оценках выполненных студентами работ.

Обратите внимание, что каждый ЭЖД изменяется, как после завершения занятия, так и после оценивания отчётов, циклически изменяясь от занятия к занятию.

3.3.3 Декомпозиция процессов учебной задачи

Примечание — **Декомпозиция блока А0 является обязательной**, а декомпозиция блоков последующих диаграмм выполняется по необходимости.

Представленные на рисунке 3.25 блоки диаграммы А0 соответствуют функциональному **уровню процессов**. Поэтому далее, декомпозиция любого из этих блоков будет представлять диаграмму, блоки которой будут соответствовать, например, функциональному уровню **операций** или **действий**. Это зависит от сложности проектируемой системы.

Для учебной цели данного пункта, проведём декомпозицию только блока А2, который наиболее важен для реализации ИС ЭЖР. Этот блок автоматизирует процесс проведения занятия преподавателем и связан с достижением основной целевой функции ИС ЭЖР — повышение производительности труда преподавателя.

Что касается семантики функционирования блока А2, то здесь достаточно чётко можно выделить три аспекта работ **уровня операций**:

- а) *аспект управления занятием*, осуществляемый преподавателем в виде доступа студентов к занятию, формулирования отдельных заданий занятия и предоставления студентам учебного материала для выполнения заданий;
- б) *выполнение последовательности заданий студентами*, оформление ими отчётов о выполненных работах и формулирование вопросов преподавателю;
- в) *аспект устранения преподавателем различных проблемных ситуаций*, связанных с выполнением студентами заданий, и ответами на все вопросы студентов.

В указанной выше проекции рассуждений, диаграмма А2 будет иметь три блока, имена которых также включаются в глоссарий проекта.

Управлять проведением занятия (А21) — *операция диаграммы А2*, выполняемая преподавателем в процессе проведения отдельного занятия и предполагающая следующее её содержание:

- а) *подключить конкретный ЭЖД* к текущему занятию для регистрации хода его выполнения;
- б) *активировать в ИС ЭЖР средства контроля доступа студентов к занятию* с целью фильтрации входного потока «**Студенты на обучение**» на выходной внешний поток «**Студенты без доступа**» и внутренний поток «**Студенты с доступом**»;
- в) *определить конкретные задания* студентам для выполнения ими учебных работ;
- г) *предоставить студентам учебный материал* для выполнения заданий.

Выполнить задание (А22) — *операция диаграммы А2*, выполняемая студентами как внутренний поток объектов — «**Студенты с доступом**». Операция осуществляется под управлением потока «**Задание**» и связана с изучением входного потока «**Учебный материал заданий**» в выходные потоки: «**Отчёты студентов**» и «**Вопросы студентов**».

Ответить на вопрос студента (А23) — *операция диаграммы А2*, выполняемая преподавателем.

давателем под управлением «УМП дисциплин». Операция преобразует «Вопросы студентов» в «Сообщения ответов».

На основании выделенных блоков, диаграмма A2 имеет вид показанный на рисунке 3.26, а глоссарий проекта дополняется определениями внутренних стрелок этой диаграммы.

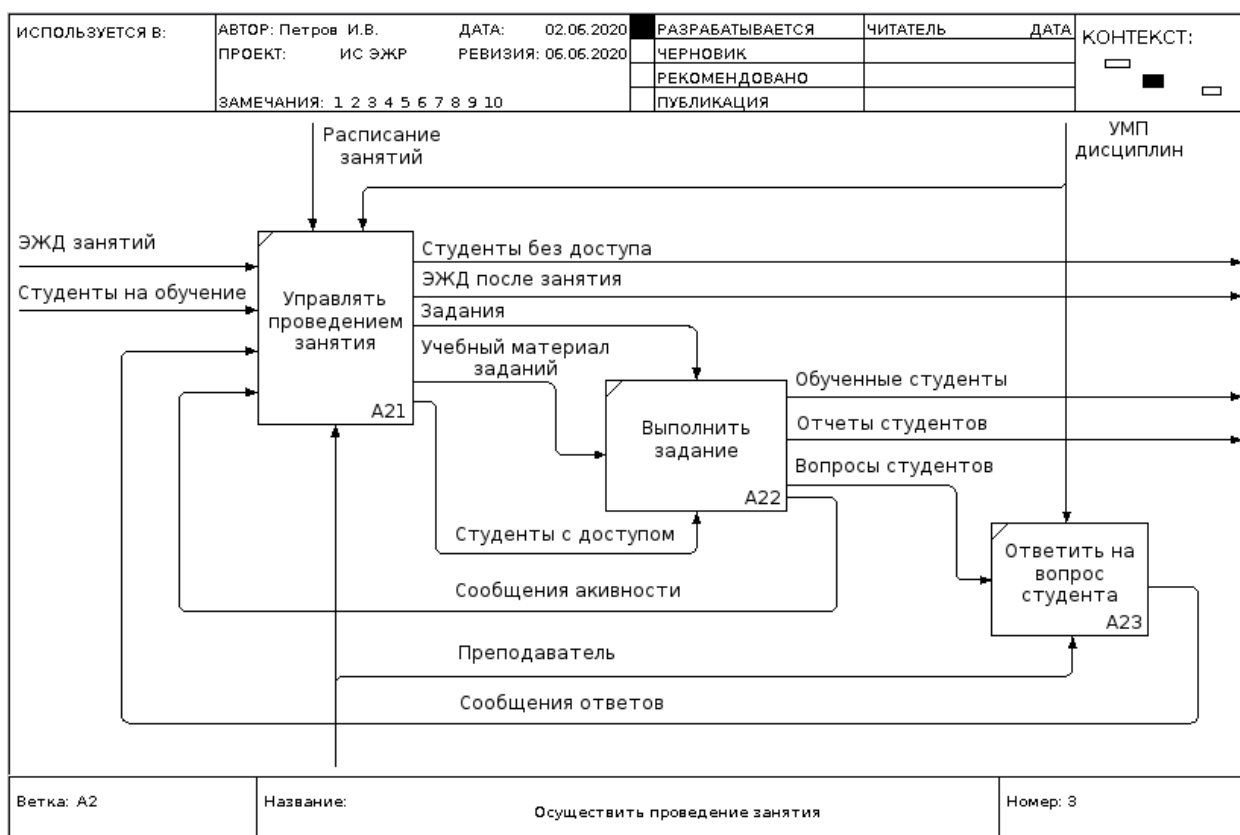


Рисунок 3.26 — Диаграмма A2, отображающая операции блока A2 ИС ЭЖР

Вопросы студентов — выходной поток блока A22 — «Выполнить задание», который регистрируется в ИС ЭЖР в виде текстовых сообщений, а затем преобразуется блоком A23 в «Сообщения ответов».

Задания — выходной поток блока A21, генерируемый преподавателем как элементарные управляющие сообщения студентам. Включает отчёты студентов по предыдущим заданиям, которые преобразуются блоком A22 в объекты «Отчёты студентов».

Сообщения активности — выходной поток блока A22, генерируемый студентами, после завершения выполнения задания или после отправки вопросов преподавателю.

Сообщения ответов — выходной поток блока A23, генерируемый преподавателем как результат текстового ответа студентам.

Студенты с доступом — выходной поток блока A21, генерируемый под надзором преподавателя как результат допуска студентов к текущему занятию, который поступает на блок A22 в качестве его «механизма».

Учебный материал заданий — выходной поток блока A21, генерируемый преподавателем для обеспечения студентов учебным материалом, включённым в «УМП дисциплин». Поступает как входной поток в блок A22.

ЭЖД после занятия — выходной поток блока A21, генерируемый преподавателем с целью сохранения всех изменений в ЭЖД.

Следует заметить, что мы провели декомпозицию только одного блока A2 диаграммы A0. Если обозначить через N_{cp} — среднее количество блоков декомпозиции на одной диа-

грамме и проводить декомпозицию ИС ЭЖР *до операций*, то *среднее количество диаграмм* D_{cp} можно определить формулой (3.1):

$$D_{cp} = N_{cp}^2 + 1 \quad (3.1)$$

Не трудно посчитать, что если $N_{cp} = 3$, то $D_{cp} = 10$.

Если декомпозицию проводить *до действий*, то получаем выражение (3.2):

$$D_{cp} = N_{cp}^3 + 1 = 28 \quad (3.2)$$

3.3.4 Декомпозиция операций учебной задачи

Примечание — **Уровень декомпозиции блоков диаграмм** определяется постановкой задачи и целью, которую должна достигать проектируемая ИС ЭЖР.

Формула (3.1), приведённая в предыдущем пункте, показывает, что, даже при минимальном значении количества блоков на каждой отдельной диаграмме, общее количество необходимых диаграмм — достаточно велико. Но, если мы учтём целевое назначение стадии концептуального проектирования — написание ТТЗ и ТЗ на целевую систему, то необходимость декомпозиции некоторых блоков окажется под вопросом.

На примере диаграммы А2 (см. рисунок 3.26), отображающей *операции блока А2 (Осуществить проведение занятия)*, мы видим следующую семантику блоков:

- а) **блок А21 (Управлять проведением занятия)** — набор согласованных действий, выполняемых преподавателем во время проведения отдельного занятия (базовый вид занятия — лабораторная работа); набор и количественный объем действий этого набора порождают проблемы, описанные в постановке учебной задачи (подраздел 2.1);
- б) **блок А22 (Выполнить задание)** — набор действий, выполняемых студентами на отдельном занятии под руководством преподавателя;
- в) **блок А23 (Ответить на вопрос студента)** — набор действий преподавателя, связанный с созданием текстового сообщения студенту.

Очевидно, что для завершения стадии концептуального проектирования ИС ЭЖР достаточно будет провести декомпозицию блока А21, чем мы и займёмся в данном пункте. Также понятно, что для раскрытия набора действий блока А21 необходимо иметь достаточный опыт проведения хотя бы лабораторных работ. Поэтому далее приводится *описание перечня автоматизируемых действий преподавателя*, без обоснования необходимости их наличия.

Подключить ЭЖД к занятию (А211) — действие, связанное с выбором необходимого для проведения занятия ЭЖД и подключения его к системе (с учётом управления «Расписание занятий»), сделав его рабочим для текущего занятия.

Активировать контроль доступа студентов (А212) — действие, связанное с запуском некоторой серверной программы, которая будет контролировать доступ к ИС ЭЖД объектов: «Студенты на обучение», «Сообщения ответов» и «Сообщения активности».

Регистрировать в ЭЖД активность студентов (А213) — действие преподавателя, связанное с просмотром очереди доступных сообщений и регистрацией в ЭЖД сообщений, заслуживающих дальнейшего внимания.

Реагировать на активность студентов (А214) — действие преподавателя, связанное с реакцией преподавателя на важные сообщения и генерацию (под управлением «УМП дис-

циплин») выходных потоков объектов: «Задания», «Учебный материал заданий» и «ЭЖД после занятия».

На основании выделенных блоков, диаграмма A21 — построена и показана на рисунке 3.27, а глоссарий проекта дополняется определениями внутренних стрелок этой диаграммы.

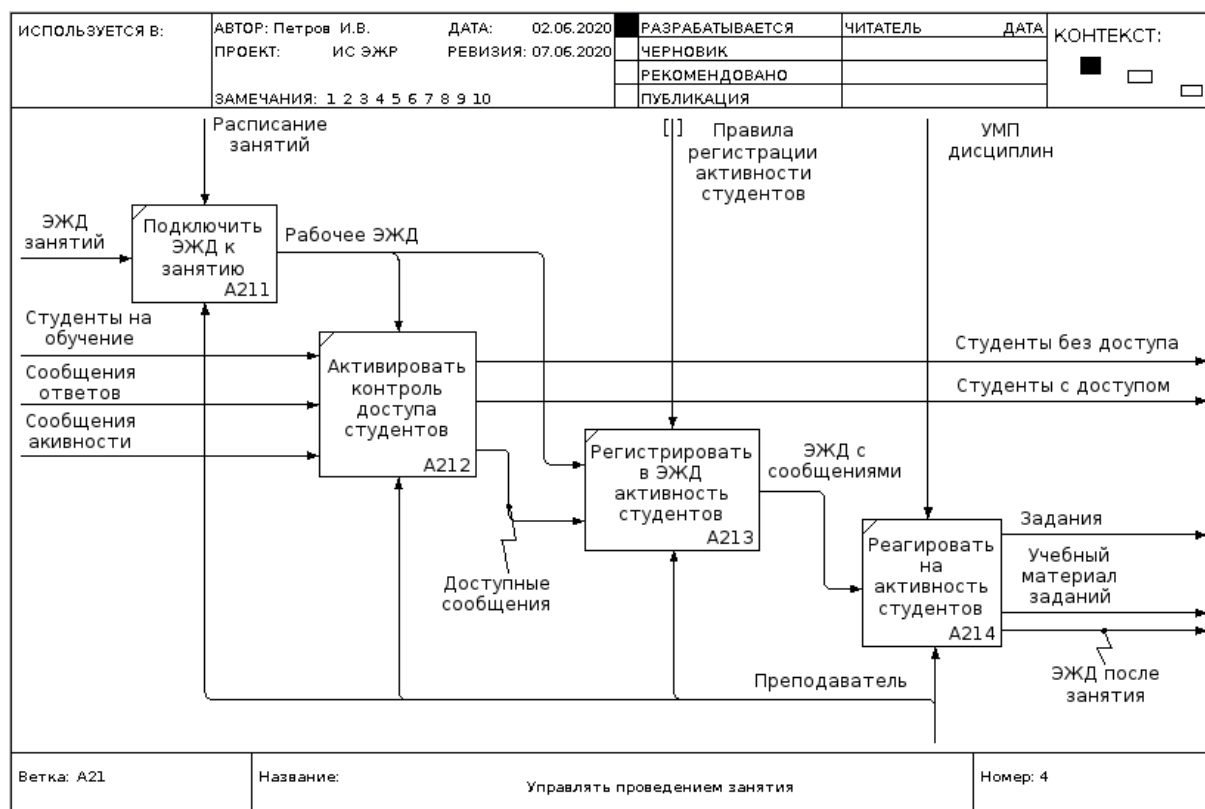


Рисунок 3.27 — Диаграмма A21, отображающая действия блока A21 ИС ЭЖР

Доступные сообщения — выходные объекты блока A212 (Активировать контроль доступа студентов), прошедшие контроль доступа к занятию.

Рабочее ЭЖД — выходной объект блока A211 (Подключить ЭЖД к занятию), управляющий контролем доступа всех объектов к занятию и являющийся входным объектом блока A213 (Регистрировать в ЭЖД активность студентов).

ЭЖД с сообщениями — выходной объект блока A213 (Регистрировать в ЭЖД активность студентов), поступающий в блок A214 (Реагировать на активность студентов), где преподаватель осуществляет оперативное управление занятием.

Правила регистрации активности студентов — входной управляющий и тунелированный поток объектов (правил), для настройки действия блока A213 (Регистрировать в ЭЖД активность студентов).

Примечание — При описании определений глоссария не следует экономить место и время на включении поясняющих записей о компонентах диаграммы.

Может показаться, что многие описания компонент диаграмм приведены излишне подробно, поскольку они достаточно хорошо видны и понятны на изображении самой диаграммы. Но следует помнить, что **глоссарий** — это справочный документ, в котором определения расположены в алфавитном порядке, поэтому любое пояснение читается и воспринимается в отрыве от других описаний компонент диаграмм проекта.

3.3.5 Формирование ТТЗ учебной задачи

Примечание — **ТТЗ** (тактико-техническое задание) является основным документом, на основании которого формируется отчёт по второй стадии проектирования («Разработка концепции ИС») и выполняется третья стадия проектирования ИС — «Техническое задание» (ТЗ).

Применительно к решаемой учебной задаче можно заметить, что рекомендуется дополнительно провести декомпозицию блоков А1 (*Подготовить список группы и ЭЖД*) и А3 (*Оценить работу студента*), но мы не будем этим заниматься, поскольку эти блоки реализуют вспомогательные функции и непосредственно не участвуют в главном процессе системы, который реализуется блоком А2 (*Осуществить проведение занятия*).

В целом, проведённой декомпозиции ИС ЭЖР, представленной совокупностью диаграмм А-0, А0, А2 и А21 (см. рисунки 3.24-3.27), уже достаточно, чтобы написать ТТЗ на проектируемую систему и завершить вторую стадию проектирования ИС. Структура и правила оформления отчёта по второй стадии такие же, как и при оформлении отчёта по первой стадии (см. раздел 2, пункт 2.5.2). Они должны соответствовать требованиям ГОСТ 7.32-2017 [34]. Титульный лист отчёта необходимо согласовать с руководителем практики от университета, а содержание основной части отчёта по НИР должно соответствовать международному стандарту РД 50-34.698-90 [33], где в Приложении 1 рекомендована следующая структура:

- 1) описание результатов изучения объекта автоматизации;
- 2) описание и оценка преимуществ и недостатков разработанных альтернативных вариантов концепции создания АС;
- 3) сопоставительный анализ требований пользователя к АС и вариантов концепции АС на предмет удовлетворения требованиям пользователя;
- 4) обоснование выбора оптимального варианта концепции и описание предлагаемой АС;
- 5) ожидаемые результаты и эффективность реализации выбранного варианта концепции АС;
- 6) ориентировочный план реализации выбранного варианта концепции АС;
- 7) необходимые затраты ресурсов на разработку, ввод в действие и обеспечение функционирования;
- 8) требования, гарантирующие качество АС;
- 9) условия приёмки системы.

Примечание — На практике, **ТТЗ** часто является основной частью более развёрнутого документа — **ТКП** (Технико-Коммерческого Предложения). Это — конкурсный документ, участвующий с ТКП других потенциальных исполнителей в отборе наиболее подходящих и привлекательных решений для заказчика ИС.

В целом, организация и проведение конкурсных мероприятий по выбору лучшего концептуального проекта ИС может потребовать множества дополнительных документов, определяемых соответствующей конкурсной комиссией. Возможны также дополнительные требования и к оформлению ТКП. Все эти вопросы выходят за рамки данного учебного пособия и далее не обсуждаются. Мы ограничиваемся тем, что ТТЗ (тактико-техническое задание) является завершающим документом учебной практики студентов, а отчёт по практике, основная часть которого содержит представленные выше разделы, является основным документом, полностью описывающим концепцию будущей ИС.

Что касается нашей учебной задачи, то основной исполнитель (Петров И.В.), заверша-

ет производственную практику полным концептуальным описанием ИС ЭЖР, используя материалы отчёта по первой стадии проектирования и ссылаясь на этот отчёт по мере необходимости. Все вопросы по окончательному оформлению и защите учебной практики Петровым И.В. также выходят за рамки данного учебного пособия, но если он полностью придерживался указанным выше требованиям, то окончательный результат должен быть положительным.

3.3.6 Техническое задание на ИС учебной задачи

Техническое задание (ТЗ) на проектируемую ИС является основным документом, который формально (юридически) закрепляет требования к проектируемой ИС.

Согласно ГОСТ 34.601-90 [8] третья стадия проектирования АС (ИС) — «*Техническое задание*» содержит только один этап — «*Разработка и утверждение задания на создание АС*». Он выполняется на основе проектного материала первых двух стадий и в качестве результата создаёт два юридически оформленных и утверждённых (подписанных заказчиком и исполнителем) документа:

- а) ***собственно документ ТЗ***, содержащий технические требования к создаваемой ИС;
- б) ***договор на создание ИС***, содержащий обязательства сторон по срокам и этапам выполняемых работ, срокам и порядку оплаты выполненных работ, а также условия завершения работ и порядок устранения разногласий между заказчиком и исполнителем работ.

Примечание — Существующая практика обучения студентов уровня бакалавриата не предполагает создания полноценного ТЗ на проектируемые ИС.

В примере задания на преддипломную практику, выданного Петрову И.В. (см. раздел 2, пункт 2.1.2, рисунок 2.1), указаны только название индивидуального задания и время прохождения практики. Даже в заданиях на ВКР (выпускные квалификационные работы), выполняемых в соответствии с документом «*Образовательный стандарт вуза ОС ТУСУР 01-2013*» [36], «*Задание на бакалаврскую работу*» ограничено требованиями двух страниц текста и содержит только шесть информационных частей:

- 1) тема бакалаврской работы (БР);
- 2) срок выдачи студентом законченной БР;
- 3) исходные данные к работе;
- 4) содержание расчётно-пояснительной записки/перечень подлежащих разработке вопросов;
- 5) перечень графического материала (с точным указанием обязательных листов презентации);
- 6) дата выдачи задания.

Неудивительно, что студенты, опираясь на указанные выше исходные данные, не имеют адекватных представлений о ТЗ и его истинном назначении.

Цель дальнейшего учебного материала данного пункта — дать общее представление о документе ТЗ и указать на важные правила его оформления.

Техническое задание (ТЗ) на автоматизированную систему (АС, ИС или АИС) оформляется в соответствии с требованиями ГОСТ 34.602-89 [9].

ТЗ является достаточно сложным документом, который отражает все стадии создания системы, начиная со стадии «*Эскизный проект*». Мифы о сложности этого документа несколько преувеличены и основаны на существующей неоднозначности понимания формулировок самого стандарта.

Как целостный документ, ТЗ на АС содержит следующие шесть частей:

- 1) лист утверждения (ЛУ);
- 2) титульный лист;
- 3) содержание;
- 4) основная часть ТЗ;
- 5) последний лист;
- 6) приложения (по необходимости).

Основная часть ТЗ на АС содержит следующие девять разделов, которые допускают деление на подразделы, пункты и подпункты:

- 1) общие сведения;
- 2) назначение и цели создания (развития) системы;
- 3) характеристика объектов автоматизации;
- 4) требования к системе;
- 5) состав и содержание работ по созданию системы;
- 6) порядок контроля и приёмки системы;
- 7) требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- 8) требования к документированию;
- 9) источники разработки.

Примечание — **Первая проблема** формирования ТЗ на проектируемому ИС — оформление «Листа утверждения» (ЛУ).

ГОСТ 34.602-89 [9] не требует обязательного наличия листа утверждения в составе ТЗ, тем не менее, многие организации-разработчики считают обязательным его присутствие. Более того, многие документы на ИС имеют такие листы, поэтому студент должен иметь общее представление о его назначении и правилах оформления.

В целом, лист утверждения оформляется на одном или нескольких листах, содержит указание о разработчике документа, названии документа, его идентификации, а также утверждающие и согласующие подписи. Пример шаблона такого листа (ЛУ) для документа ТЗ показан на рисунке 3.28.

Прежде всего обратим внимание, что левая сторона листа утверждения отведена для утверждающих и согласующих подписей заказчика системы и других согласующих организаций, а правая сторона — для организации-разработчика (исполнителя). Сам этот лист хранится у разработчика и, в случае необходимости, его копия передаётся заказчику или другим уполномоченным сторонам.

В середине листа утверждения находятся надписи, идентифицирующие документ, а также надпись обозначения документа, которая в соответствии с требованиями ГОСТ 34.201-89 [37] имеет вид, показанный на рисунке 3.29.

Здесь «*Код документа*» — двухбуквенное обозначение, которое для технического задания равно «ТЗ», а «*Обозначение системы (части системы)*», согласно этому же источнику

[37], показано на рисунке 3.30.

Наименование организации-разработчика							
УТВЕРЖДАЮ				УТВЕРЖДАЮ			
Руководитель (должность, наименование организации- заказчика АСУ)				Руководитель (должность, наименование организации- разработчика АСУ)			
Личная подпись		Расшифровка подписи		Личная подпись		Расшифровка подписи	
Дата		М.П.		Дата		М.П.	
Наименование вида АС Наименование объекта автоматизации Сокращенное наименование АС ТЕХНИЧЕСКОЕ ЗАДАНИЕ Лист утверждения XXXXXXXX.XXXXXXXX.XXX.ТЗ-ЛУ							
СОГЛАСОВАНО				СОГЛАСОВАНО			
(должность)				(должность)			
(ФИО)				(ФИО)			
«__» _____ 20__ г.				«__» _____ 20__ г.			
М.П.				М.П.			
СОГЛАСОВАНО				СОГЛАСОВАНО			
(должность)				(должность)			
(ФИО)				(ФИО)			
«__» _____ 20__ г.				«__» _____ 20__ г.			
М.П.				М.П.			

Рисунок 3.28 — Шаблон листа утверждения для документа ТЗ

	XX.	XX.	X-	X.	М
Обозначение системы (части системы)					
Код документа					
Порядковый номер документа одного наименования					
Номер редакции документа					
Номер части документа					
Признак документа, выполненного на машинных носителях					

Рисунок 3.29 — Обозначение документа по ГОСТ 34.201-89 [37]

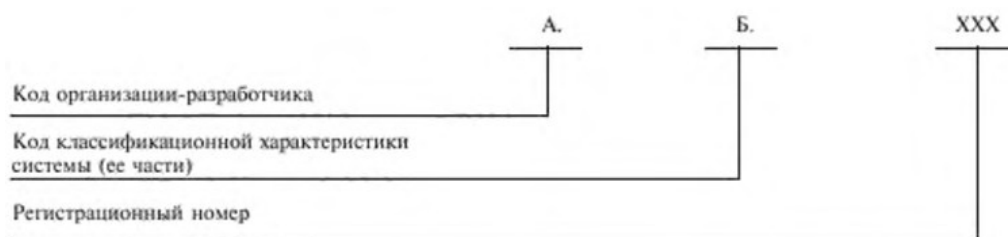


Рисунок 3.30 — Обозначение системы или её части по ГОСТ 34.201-89 [37]

Код организации-разработчика присваивается общесоюзным классификатором для предприятий, учреждений и организаций (ОКПО). Для ТУСУР это значение равно **02069326**.

Код классификационной характеристики системы (её части) присваивают в соответствии с правилами, установленными в отрасли (например, ОКП на основе подкласса 425000 «Программно-технические комплексы для автоматизированных систем» или 507000 «Прикладные программные средства учебного назначения», см. [38]) или на основе общесоюзного классификатора подсистем и комплексов задач АСУ, который в настоящее время не поддерживается.

Регистрационный номер — номер организации-разработчика (от 001 до 999), под которым она регистрирует проектируемую систему.

Примечание — На практике, «*Обозначение системы (части системы)*» — это всего лишь идентификатор проектируемой системы для хранения и извлечения документации организацией разработчиком.

Каждая организация-разработчик обычно имеет свой классификатор, которым и должен пользоваться студент. Например, образовательный стандарт ТУСУР [36] требует следующую структуру идентификатора, показанную на рисунке 3.31. При этом:

- в качестве **кода разработчика** (в учебных проектах) рекомендуется использовать *аббревиатуру выпускающей кафедры*;
- порядковый регистрационный номер разработки** или номер редакции документа *назначается по указаниям кафедры, организующей проектирование*.



Рисунок 3.31 — Обозначение проектов по ОС ТУСУР 01-2013 [36]

Согласно требованиям кодирования, представленным рисунком 3.31 и организационным данным учебной задачи (см. далее по тексту таблицу 3.2), лист утверждения ТЗ примет вид, показанный на рисунке 3.32.

Теперь, сам документ ТЗ, его титульный лист и завершающий лист будут иметь шифр **АСУ.507200.001 ТЗ 01** и по этому шифру заказчик может его заказывать.

Пример титульного листа ТЗ для учебной задачи показан на рисунке 3.33. Далее, на рисунке 3.34 показан завершающий лист ТЗ.

Таблица 3.2 — Организационные данные учебной задачи

Поле идентификатора	Значение
Код разработчика	АСУ
Код ОКП	507200 - «Программные средства для управления учебным процессом» [38]
Номер регистрации	001 (считаем, что под таким номером наша система зарегистрирована секретарём кафедры АСУ)
Код вида документа	ТЗ-ЛУ
Номер документа	01
Лица утверждающие	Зав.кафедрой — Романенко В.В. (от заказчика) Руководитель — Резник В.Г. (от разработчика)
Лица согласовывающие	Руководитель практики — Григорьева М.В. (от заказчика) Студент — Петров И.В. (от разработчика)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР) Кафедра автоматизированных систем управления (АСУ)	
<p style="text-align: center;">УТВЕРЖДАЮ</p> <p>Зав.кафедрой АСУ, доцент</p> <p>_____ В.В. Романенко</p> <p>«__» _____ 20__ г.</p> <p style="text-align: center;">М.П.</p>	<p style="text-align: center;">УТВЕРЖДАЮ</p> <p>Научный руководитель, доцент кафедры АСУ</p> <p>_____ В.Г. Резник</p> <p>«__» _____ 20__ г.</p> <p style="text-align: center;">М.П.</p>
Автоматизированная информационная система (АИС) Электронный журнал руководителя АИС ЭЖР ТЕХНИЧЕСКОЕ ЗАДАНИЕ Лист утверждения АСУ.507200.001 ТЗ-ЛУ 01	
<p style="text-align: center;">СОГЛАСОВАНО</p> <p>Руководитель учебной практики, доцент</p> <p>_____ Григорьева М.В.</p> <p>«__» _____ 20__ г.</p> <p style="text-align: center;">М.П.</p>	<p style="text-align: center;">УТВЕРЖДАЮ</p> <p>Исполнитель, студент группы 447-1 кафедры АСУ</p> <p>_____ Петров И.В.</p> <p>«__» _____ 20__ г.</p> <p style="text-align: center;">М.П.</p>

Рисунок 3.32 — Лист утверждения ТЗ для учебной задачи

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

УТВЕРЖДЕН
АСУ.507200.001 ТЗ-ЛУ 01

**Автоматизированная информационная система (АИС)
Электронный журнал руководителя
АИС ЭЖР
ТЕХНИЧЕСКОЕ ЗАДАНИЕ
АСУ.507200.001 ТЗ-ЛУ 01
На 95 листах**

Действует с « 20 » _____ мая _____ 2020 года

Рисунок 3.33 — Титульный лист ТЗ для учебной задачи

АСУ.507200.001 ТЗ 01

95

СОСТАВИЛИ

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпись	Дата
Кафедра АСУ ТУСУР	студент	Петров И.В.		19.05.20

СОГЛАСОВАНО

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпись	Дата
Кафедра АСУ ТУСУР	доцент	Григорьева М.В.		19.05.20

Рисунок 3.34 — Завершающий лист ТЗ для учебной задачи

Как отмечено выше, ГОСТ 34.602-89 [9] требует изложения содержания ТЗ в виде девяти обязательных разделов. Полный перечень всех разделов, подразделов и пунктов ТЗ при-

ведён ниже мелким шрифтом, как справочная выдержка из источника [9]:

- 1.1 Полное наименование системы и ее условное обозначение.
- 1.2 Шифр темы или шифр (номер) договора.
- 1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты.
- 1.4 Перечень документов, на основании которых создаётся система, кем и когда утверждены эти документы.
- 1.5 Плановые сроки начала и окончания работы по созданию системы.
- 1.6 Сведения об источниках и порядке финансирования работ.
- 1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы.
- 2 Назначение и цели создания системы.
 - 2.1 Назначение системы.
 - 2.2 Цели создания системы.
- 3 Характеристика объектов автоматизации.
 - 3.1 Краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию.
 - 3.2 Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.
- 4 Требования к системе.
 - 4.1 Требования к системе в целом.
 - 4.1.1 Требования к структуре и функционированию системы.
 - 4.1.1.1 Перечень подсистем, их назначение, основные характеристики, требования к числу уровней иерархии и степени централизации системы.
 - 4.1.1.2 Требования к способам и средствам связи для информационного обмена между компонентами системы.
 - 4.1.1.3 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости.
 - 4.1.1.4 Требования к режимам функционирования системы.
 - 4.1.1.5 Требования по диагностированию системы.
 - 4.1.1.6 Перспективы развития, модернизации системы.
 - 4.1.2 Требования к численности и квалификации персонала системы и режиму его работы.
 - 4.1.2.1 Требования к численности персонала (пользователей) АС.
 - 4.1.2.2 Требования к квалификации персонала, порядку его подготовки и контроля знаний и навыков.
 - 4.1.2.3 Требуемый режим работы персонала АС.
 - 4.1.3 Показатели назначения.
 - 4.1.3.1 Степень приспособляемости системы к изменению процессов и методов управления к отклонению параметров объекта управления.
 - 4.1.3.2 Допустимые пределы модернизации и развития системы.
 - 4.1.3.3 Вероятностно-временные характеристики, при которых сохраняется целевое назначение системы.
 - 4.1.4 Требования к надёжности.
 - 4.1.4.1 Состав и количественные значения показателей надёжности для системы в целом или ее подсистем.

4.1.4.2 Перечень аварийных ситуаций, по которым должно быть регламентированы требования к надёжности, и значения соответствующих показателей.

4.1.4.3 Требования к надёжности технических средств и программного обеспечения.

4.1.4.4 Требования к методам оценки и контроля показателей надёжности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.

4.1.5 Требования к безопасности.

4.1.6 Требования к эргономике и технической эстетике.

4.1.7 Требования к транспортабельности для подвижных АС.

4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы.

4.1.8.1 Условия и регламент (режим) эксплуатации, которые должны обеспечивать использование технических средств (ТС) системы с заданными техническими показателями, в том числе виды и периодичности обслуживания ТС системы или допустимость работы без обслуживания.

4.1.8.2 Предварительные требования к допустимым площадям для размещения персонала и ТС системы, к параметрам сетей энергоснабжения.

4.1.8.3 Требования по количеству, квалификации обслуживающего персонала и режимам его работы.

4.1.8.4 Требования к составу, размещению и условиям хранения комплекта запасных изделий и приборов.

4.1.8.5 Требования к регламенту обслуживания.

4.1.9 Требования к защите информации от несанкционированного доступа.

4.1.10 Требования по сохранности информации при авариях.

4.1.11 Требования к средствам защиты от влияния внешних воздействий.

4.1.11.1 Требования к радиоэлектронной защите средств АС.

4.1.11.2 Требования по стойкости, устойчивости и прочности к внешним воздействиям (среде применения).

4.1.12 Требования к патентной чистоте.

4.1.13 Требования по стандартизации и унификации.

4.1.14 Дополнительные требования.

4.1.14.1 Требования к оснащению системы устройствами для обучения персонала (тренажёрами, другими устройствами аналогичного назначения) и документацией на них.

4.1.14.2 Требования к сервисной аппаратуре, стендам для проверки элементов системы.

4.1.14.3 Требования к системе, связанные с особыми условиями эксплуатации.

4.1.14.4 Специальные требования по усмотрению разработчика или заказчика системы.

4.2 Требования к функциям (задачам), выполняемым системой.

4.2.1 Требования к подсистеме. Перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации.

4.2.2 Временной регламент реализации каждой функции, задачи (или комплекса задач).

4.2.3 Требования к качеству реализации каждой функции (задачи или комплекса задач), к форме представления выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов.

4.2.4 Перечень и критерии отказов для каждой функции, по которой задаются требования по надёжности.

4.3 Требования к видам обеспечения.

4.3.1 Требования к математическому обеспечению.

4.3.2 Требования к информационному обеспечению.

- 4.3.2.1 Требования к составу, структуре и способам организации данных в системе.
- 4.3.2.2 Требования к информационному обмену между компонентами системы.
- 4.3.2.3 Требования к информационной совместимости со смежными системами.
- 4.3.2.4 Требования по использованию общесоюзных и зарегистрированных республиканских, отраслевых классификаторов, унифицированных документов и классификаторов, действующих на данном предприятии
- 4.3.2.5 Требования по применению систем управления базами данных.
- 4.3.2.6 Требования к структуре процесса сбора, обработки, передачи данных в системе и представлению данных.
- 4.3.2.7 Требования к защите данных от разрушений при авариях и сбоях в электропитании системы.
- 4.3.2.8 Требования к контролю, хранению, обновлению и восстановлению данных.
- 4.3.2.9 Требования к процедуре придания юридической силы документам, производимым техническими средствами АС (в соответствии с ГОСТ 6.10.4).
- 4.3.3 Требования к лингвистическому обеспечению.
- 4.3.4 Требования к программному обеспечению.
- 4.3.4.1 Требования к независимости программных средств от используемых СBT и операционной среды.
- 4.3.4.2 Требования к качеству программных средств, а также к способам его обеспечения и контроля.
- 4.3.4.3 Требования по необходимости согласования вновь разрабатываемых программных средств с фондом алгоритмов и программ.
- 4.3.5 Требования к техническому обеспечению.
- 4.3.5.1 Требования к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию в системе.
- 4.3.5.2 Требования к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.
- 4.3.6 Требования к метрологическому обеспечению.
- 4.3.6.1 Предварительный перечень измерительных каналов.
- 4.3.6.2 Требования к точности измерений параметров и (или) к метрологическим характеристикам измерительных каналов.
- 4.3.6.3 Требования к метрологической совместимости технических средств системы.
- 4.3.6.4 Перечень управляющих и вычислительных каналов системы, для которых необходимо оценивать точностные характеристики.
- 4.3.6.5 Требования к метрологическому обеспечению технических и программных средств, входящих в состав измерительных каналов системы, средств встроенного контроля, метрологической пригодности измерительных каналов и средств измерений, используемых при наладке и испытаниях системы.
- 4.3.6.6 Вид метрологической аттестации (государственная или ведомственная) с указанием порядка ее выполнения и организаций, проводящих аттестацию.
- 4.3.7 Требования к организационному обеспечению.
- 4.3.7.1 Требования к структуре и функциям подразделений, участвующих в функционировании системы или обеспечивающих эксплуатацию.
- 4.3.7.2 Требования к организации функционирования системы и порядку взаимодействия персонала АС и персонала объекта автоматизации.
- 4.3.7.3 Требования к защите от ошибочных действий персонала системы.
- 4.3.8 Требования к методическому обеспечению.
- 4.3.9 Требования к другим видам обеспечения системы.
- 5 Состав и содержание работ по созданию системы.

5.1 Перечень документов по ГОСТ 34.201, предъявляемых по окончании соответствующих стадий и этапов работ.

5.2 Вид и порядок проведения экспертизы технической документации (стадия, этап, объем проверяемой документации, организация-эксперт).

5.3 Программа работ, направленных на обеспечение требуемого уровня надёжности разрабатываемой системы.

5.4 Перечень работ по метрологическому обеспечению на всех стадиях создания системы с указанием их сроков выполнения и организации-исполнителей.

6 Порядок контроля и приёмки системы.

6.1 Виды, состав, объем и методы испытаний системы и ее составных частей.

6.2 Общие требования к приёмке работ по стадиям.

6.3 Статус приёмочной комиссии.

7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

7.1 Приведение поступающей в систему информации (в соответствии с требованиями к информационному и лингвистическому обеспечению) к виду, пригодному для обработки с помощью ЭВМ.

7.2 Изменения, которые необходимо осуществить в объекте автоматизации.

7.3 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ.

7.4 Создание необходимых для функционирования системы подразделений и служб.

7.5 Сроки и порядок комплектования штатов и обучения персонала.

8 Требования к документированию.

8.1 Согласованный разработчиком и заказчиком системы перечень подлежащих разработке комплектов и видов документов, соответствующих требованиям ГОСТ 34.201 и НТД отрасли заказчика; перечень документов, выпускаемых на машинных носителях; требования к микрофильмированию документации.

8.2 Требования по документированию комплектующих элементов межотраслевого применения в соответствии с требованиями ЕСКД и ЕСПД.

8.3 При отсутствии государственных стандартов, определяющих требования к документированию элементов системы, дополнительно включают требования к составу и содержанию таких документов.

9 Источники разработки.

Перечень принятых сокращений.

Примечание — Практика разработки, согласования и утверждения ТЗ на АС допускает использование только части подразделов и пунктов содержания, представленного выше.

Представленная выше структура текста ТЗ является *рекомендованной и допускает исключение подразделов и пунктов* мало относящихся к сущности проектируемой системы. Более того, допускается, по согласованию сторон, включать новые разделы подразделы, которые раскрывают или уточняют сущность решаемых системой задач.

В целом, ТЗ является сложным и достаточно объёмным документом, структура которого должна быть изучена и понятна студенту.

3.3.7 Итог применения методологии IDEF0 к проектированию ИС

Рассмотренная в данном подразделе методология IDEF0 для концептуального проектирования учебной задачи и формирования её ТЗ наглядно показывает всю сложность принятия проектных решений и оформления соответствующей документации. Более сложные задачи могут потребовать построение дополнительных поясняющих моделей, которые не реали-

зуются средствами изученной методологии структурного проектирования.

Анализ диаграмм, представленных ранее на рисунках 3.24 — 3.27, показывает, что *не всегда понятно* в какой последовательности должны выполняться объявленные и задокументированные функции. Особенно это актуально для *действий*, представленных и описанных на нижних уровнях декомпозиции диаграмм IDEF0.

Для построения дополнительных поясняющих моделей концептуального проектирования рекомендуется использовать методологию IDEF3.

3.4 Моделирование потоков работ IDEF3

Вторым по значимости методологией структурного моделирования бизнес-процессов SADT является стандарт IDEF3.

IDEF3 (Integrated DEFinition for Process Description Capture Method) — методология структурного моделирования и стандарт документирования процессов, происходящих в системе, показывающий причинно-следственные связи между ситуациями и событиями в понятной экспертам форме и использующий структурный метод выражения знаний о том, как функционирует система, процесс или предприятие.

Формальное описание этого стандарта доступна в англоязычной публикации [39], содержащей 236 страниц текста и иллюстраций, которая находится в свободном доступе по адресу: <http://www.staratel.com/iso/IDEF/IDEF3/Idef3.pdf>. Этой публикации мы и будем придерживаться, по крайней мере, в плане демонстрации рисунков. Кроме того, на просторах интернета имеются многочисленные частные публикации и трактовки этого документа, содержащие как правило методические рекомендации по его применению. В частности, краткий обзор всех стандартов IDEF содержится в учебном пособии Цукановой О.А. [40], который рекомендуется студентам для самостоятельного изучения.

Примечание — Различные методические публикации по методологии IDEF3 делают акцент на использовании различных терминов, что затрудняет единое понимание и последующее практическое использование этой методологии.

Как и любая структурная модель, диаграммы стандарта IDEF3 содержат *узлы* и *связи* между ними, которые имеют достаточно большой набор интерпретаций, что увеличивает выразительные возможности графического языка, но затрудняет последующее понимание семантики создаваемых диаграмм моделей. Более того, IDEF3 состоит из двух методов:

- а) **Process Flow Description (PFD)** — Описание технологических процессов, с указанием того, что происходит на каждом этапе технологического процесса;
- б) **Object State Transition Description (OSTD)** — описание переходов состояний объектов, с указанием того, какие существуют промежуточные состояния у объектов в моделируемой системе.

Основу методологии IDEF3 составляет графический язык описания процессов. Модель в нотации IDEF3 может содержать два типа диаграмм:

- а) **Описание Последовательности Этапов Процесса** — Process Flow Description Diagrams (PFDD);
- б) **Сети Трансформаций Состояния Объекта** — Object State Transition Network (OSTN);
- в) **Комбинированный вариант**, использующий графические элементы обоих предыдущих типов диаграмм.

Чтобы избежать различных неоднозначных толкований изучаемой методологии, ограничим изучение методологии IDEF3 только следующими четырьмя пунктами, достаточными для применения этой методологии в задачах концептуального проектирования ИС.

3.4.1 Элементы графического языка описания процессов IDEF3

Основным мотивом разработки методологии IDEF3 была необходимость предоставить концепции, синтаксис и процедуры для создания описаний системных требований к процессам. Эти требования подразумевают, что IDEF3, как метод, должен поддерживать описания следующих элементов:

- 1) *сценарии* организационной деятельности;
- 2) *роли типов пользователей* в этих организационных действиях.
- 3) *пользовательские сценарии* или взаимодействие пользователя с информационной системой на пользовательско-функциональном уровне;
- 4) *реакция системы* на пользовательские функции.
- 5) *пользовательские классы* и разграничение пользовательских классов.
- 6) *декларации сроков*, последовательности и ограничений ресурсов.
- 7) *объекты пользовательского интерфейса*, например, меню, ключевые слова, экраны и дисплеи.

На рисунке 3.35 показаны элементы графического языка IDEF3, которые должны обеспечить отображение заявленных описаний процессов.

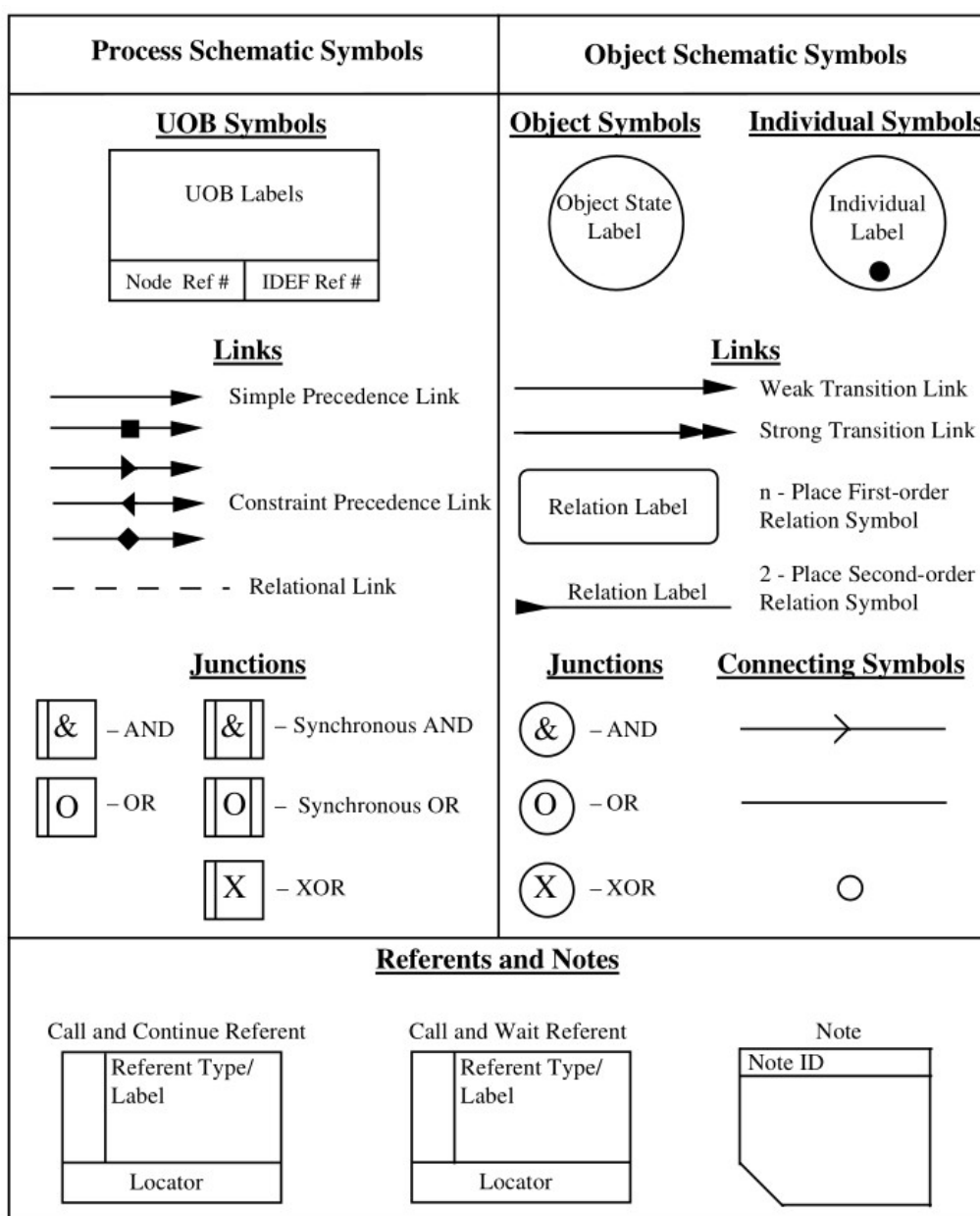


Рисунок 3.35 — Элементы языка IDEF3 [39]

Верхняя часть рисунка 3.35 разделена на две части:

- а) **Process Schematics Symbols** — графические символы предназначенные для отображения схем (сценариев, диаграмм) *процессов*, таких как блоки **UOB** (*Units of Behavior, Единицы Поведения*), стрелки связей блоков UOB (Links) и блоки перекрёстков (Junctions);
- б) **Object Schematics Symbols** — аналогичные символы для схем состояний объектов.

В нижней части рисунка 3.35 представлены вспомогательные графические символы, названные как *Референты* (Referents) и *Замечания* (Notes).

Notes — графические элементы *Замечания* (*Примечания*) используемые для обеспечения дополнительной информации в процессе моделирования, которые могут присоединяться к диаграммам иллюстраций, текста, комментариев и другим элементам для предоставления возможности выразить идеи или концепции вместо использования относительных связей.

Referents (*Референты*) — типизированные и именованные объекты ссылок, привязанные к UOB или перекрёсткам для описания особых типов связи, которые нецелесообразно указывать ссылками *Links*.

Референты могут использоваться в схемах процессов и объектов IDEF3 для следующих целей:

- а) для ссылки на ранее определённый блок UOB без дублирования его определения, чтобы указать, что другой экземпляр ранее определённого блока UOB возникает в определённой точке процесса;
- б) для передачи управления или указания обратной связи в обработке других UOB.
- в) для формирования ссылок или связей между схемами процессов и схемами объектов.

Референты являются активными во время активности графического элемента, к которому они подсоединены, а по цели воздействия на ссылаемый элемент диаграммы (схемы) подразделяются на два типа:

- а) **Call and Continue Referent** — референт *Вызывать-и-Продолжить только иницирует ссылочный элемент*, что должно произойти до завершения его собственной активности;
- б) **Call and Wait Referent** — референт *Вызывать-и-Ожидать иницирует ссылочный элемент и ждёт завершения его функционирования*, что должно произойти до завершения собственной активности референта.

В таблице 3.3 представлены типы референтов и обозначения их меток, используемые в диаграммах методологии IDEF3.

Таблица 3.3 — Типы и обозначения меток референтов методологии IDEF3

Типы референта	Обозначение меток референта
UOB	Имя функционального элемента UOB (№ UOB)
SCENARIO	Название сценария (№ Scenario)
TS (Transaction Schematic)	Название диаграммы перехода состояний (№ диаграммы перехода)
GO-TO	Имя функционального элемента UOB (№ UOB, № сценария или декомпозиции, в которой находится элемент)

Примечание — *Референты* и *Замечания* улучшают понимание диаграмм IDEF3, придают им дополнительный смысл и упрощают их построение, поскольку предназначены для «минимизации беспорядка» как в схемах (диаграммах) процессов, так и схемах объектов.

Уже перечисленные элементы графического языка IDEF3 показывают, что диаграммы, построенные по методологии IDEF3 обладают достаточно большими средствами выразительности, далеко выходящими за рамки проектирования только информационных систем (ИС).

Главная особенность методологии IDEF3, отличающая её от методологии IDEF0, — возможность отображать временные последовательности выполнения процессов, работ, событий и других элементов действий, отображаемых графическими элементами *UOB*, *Object State Label* и *Junctions*.

Указанная особенность методологии IDEF3 сосредоточена и раскрывается метаопределением понятия *Сценарии*. Чтобы наиболее точно раскрыть это понятие, приведу прямой текст оригинала [39, стр. 10-12]: «Понятие сценария или истории используется в качестве базовой организационной структуры для описаний процессов IDEF3. Сценарий можно рассматривать как повторяющуюся ситуацию, набор ситуаций, описывающих типичный класс проблем, решаемых организацией или системой, или обстановку, в которой происходит процесс. Сценарии устанавливают фокус и граничные условия описания. Использование сценариев таким образом использует склонность людей описывать то, что они знают, в терминах упорядоченной последовательности действий в контексте данного сценария или ситуации. Сценарии также обеспечивают удобное средство для организации коллекций знаний, ориентированных на процесс.

Поскольку основная роль сценария заключается в связывании контекста описания процесса IDEF3, важно дать ему соответствующее имя. Названия сценариев часто принимают форму повелительного наклонения (например, глагол или глагольные фразы, такие как «Оформить заказ на покупку», «Проверить пригодность» и т. д.), а иногда могут принимать форму герундия (например, глагол, который функционирует как существительное, например, «Выполнение» или «Проверки согласованности»). Хорошо подобранное название сценария гарантирует, что у пользователей описания возникнут соответствующие ассоциации с описываемыми реальными ситуациями. Правильная идентификация, характеристика и наименование сценариев является необходимым шагом к созданию ориентированных на процесс описаний процессов IDEF3. Следующие примеры являются типичными именами сценариев.

1. Разработайте дизайн штампа для боковой апертурной панели.
2. Обработка жалоб клиента.
3. Реализовать запрос на инженерное изменение.

Описание процесса IDEF3 разработано с использованием двух стратегий приобретения знаний: стратегии, ориентированной на процесс, и стратегии, ориентированной на объект. Стратегия, ориентированная на процесс, организует знания о процессах с упором на процессы и их временные, причинно-следственные и логические отношения в рамках сценария. Второе измерение организует знания о процессах, сосредоточив внимание на объектах и их поведении при изменении состояния в одном или нескольких сценариях.

Используя одну или обе эти стратегии получения знаний о процессах, пользователи IDEF3 разрабатывают описания процессов IDEF3. Обе стратегии используют базовые элементы языка IDEF3 для задания и выражения утверждений, формирующих описание.

Графические проекции информации, содержащейся в описаниях процессов, создаются с использованием графического языка IDEF3. Эти графические проекции, используемые как для непосредственной записи информации о процессе, так и в качестве механизма для отображения информации о процессе, называются схемами.

Два типа схем IDEF3 параллельны двум стратегиям получения знаний о процессах. Схема процесса IDEF3 отображает представление сценария, ориентированное на процесс. Схемы объектов поддерживают графическое отображение объектно-ориентированной информации. Схемы объектов, которые отображают объектно-центрированное представление одного сценария, называются схемами переходов. Схемы переходов, которые отображают дополнительные объекты и отношения между объектами для предоставления информации, устанавливающей контекст, называются расширенными схемами переходов. Схемы объектов, которые отображают объектно-центрированную информацию, охватывающую несколько сцена-

риев, называются просто схемами объектов.

Описание процесса IDEF3 может содержать ноль или более схем процессов и ноль или более схем объектов. Например, запись о том, что конкретный объект распознаётся участниками домена, считается частью описания этого домена. Идентифицированный таким образом объект может иметь или не иметь схему объекта, связанную с ним в описании. Тем не менее, эти объекты считаются частью описания. Концепция сценария используется для организации представлений, ориентированных как на процессы, так и на объекты. Набор сценариев и информации, предназначенная для организации представлений, и является описанием процесса в стандарте IDEF3».

Для концептуального проектирования ИС нецелесообразно использовать полную методологию стандарта IDEF3.

Можно обсуждать заявленный тезис применительно к другим стадиям проектирования ИС, но стадия концептуального проектирования требует лишь уточнения функционального описания процессов, которые полностью не раскрываются или не могут быть раскрыты средствами методологии IDEF0. Это обосновывается следующими аргументами.

Аргумент 1. Метод схем объектов стандарта IDEF3 серьёзно конкурирует с другими объектными моделями, например, стандартом UML и другими объектными технологиями, которые рассматриваются в следующем разделе данного пособия.

Аргумент 2. Находится под сомнением современная актуальность разработки инструментальных CASE-средств, ориентированных на реализацию полной методологии IDEF3. Известная инструментальная система BPWIN, поддерживающая построение диаграмм IDEF3, содержит методические описания и примеры, которые демонстрируют сценарии процессов.

Аргумент 3. Многие методические описания, демонстрирующие методику практического использования IDEF3, применяют вместо определения UOB (Unit of Behavior) определение UOW (Unit of Work), трактуя IDEF3 как методологию потоков работ, а полученные диаграммы — как WFD (Work Flow Description/Diagrams).

Следуя общей сложившейся традиции, дальнейшее изложение учебного материала данного раздела будет ориентировано на описание диаграмм потоков процессов или Process Flow Description Diagrams (PFDD).

3.4.2 Синтаксис и семантика языка IDEF3

Основным структурным функциональным элементом модели IDEF3 является блок UOB (Unit of Behavior), графический символ которого показан в левой верхней части рисунка 3.35. Этот символ обозначается *непрерывным именованным прямоугольником* и по общему семантическому назначению соответствует функциональному блоку модели IDEF0. Но поскольку диаграммы методологии IDEF3 не учитывают метапонятия связей *Управления* и *Механизма*, то отличительные особенности UOB выражаются следующими особенностями:

- а) UOB содержит только стрелки (связи) входов и выходов, определяя временную очерёдность активности UOB, а также присоединение к UOB графических элементов *Референтов* и *Замечаний*;
- б) именование UOB не поддерживает градацию его функционального содержания по уровням иерархии, давая UOB множественную интерпретацию: функция, процесс, действие, акт, событие, сценарий и другие подобные семантики.

Примечание — Казалось бы указанные выше особенности UOB упрощают сам процесс построения диаграмм методологии IDEF3 и наполняют UOB большим содержанием. Но, как уже было отмечено выше, многообразие толкований имён UOB приводит к неоднозначности последующей интерпретации полученных описаний потребителями уже созданных диаграмм.

Указанная проблема наглядно демонстрируется рисунком 3.36.

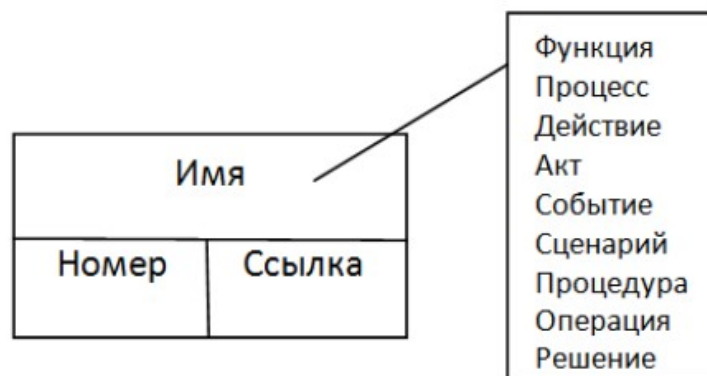


Рисунок 3.36 — Синтаксис и семантика графического элемента UOB [40]

Действительно, указание в имени UOB её функциональной интерпретации не только увеличивает длину этого имени, но и затрудняет последующее составление различных текстовых описаний получаемых диаграмм: официальной документации проектов, различных инструкций и других сопутствующих документов. Именно поэтому различные методические пособия по IDEF3 трактуют UOB как функции, действия или просто процессы. Обычно эти вариации обобщённой семантики UOB вызваны недостаточной квалификацией проектировщиков или стремлением приспособиться (угодить) ограниченным познаниям потребителей конечного продукта.

Частично указанный недостаток именования и интерпретации UOB разрешается использованием «Ссылки» на соответствующий блок диаграммы IDEF0, а также рекомендацией именовать функциональный элемент UOB активным глаголом, как это требует сама методология IDEF0.

Другой и более радикальный подход — интерпретировать функциональное содержание UOB как *Работа* и использовать это понятие при оформлении всей проектной документации. Например, авторы методического пособия [41] Уральского государственного технического университета (УГТУ), описывающего работу CASE-средств BPWIN и ERWIN, в явном виде заменяют UOB (Unit of Behavior) на UOW (Unit of Work).

Работа — синтаксическое обозначение и семантическая интерпретация функционального содержания и имени UOB, обобщающее частичные функциональные интерпретации имени UOB, показанные ранее на рисунке 3.36.

Безусловно, «механическая» подмена UOB на UOW искажает теоретическую интерпретацию стандарта IDEF3, семантически определяя её диаграммы как описание *Потока работ* (Workflow), но в плане концептуального проектирования ИС это — приемлемо, поскольку акцентирует внимание на главных свойствах проектируемых информационных систем.

В дальнейшем изложении учебного материала мы будем использовать именно термин *Работа*, учитывая его общую семантическую ограниченность, указанную выше. Именно термин *Работа* будет использоваться нами для интерпретации семантики *связей* между UOB.

Связи (Links) — важнейшие элементы любой графической диаграммы, задающие всё многообразие отношений между работами моделируемой системы.

На рисунке 3.37 показаны все виды связей стандарта IDEF3, предназначенные для процессного моделирования систем и разделённые на три группы: *связи простой очередности*, *связи приоритета ограничения* и *реляционные связи*.

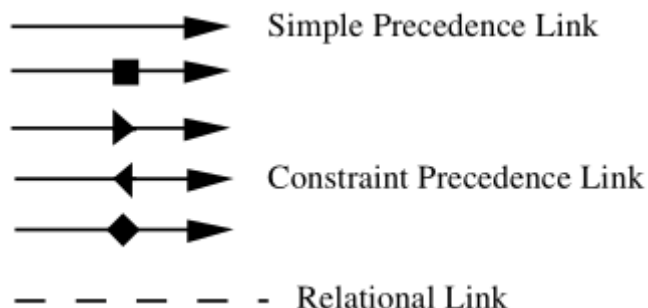


Рисунок 3.37 — Синтаксис графических элементов связей между UOB [39]

Связь простой очередности (Simple Precedence Link) — наиболее простой и используемый тип связи, изображаемый сплошной стрелкой с наконечником, которая показывает временной приоритет отношений между двумя функциональными блоками UOB.

На рисунке 3.38 показана связь простой очередности между двумя UOB с условными именами *A* и *B*. Она семантически указывает, работа UOB с именем *B* может начаться только после завершения работы *A* и возможно через некоторый интервал времени.

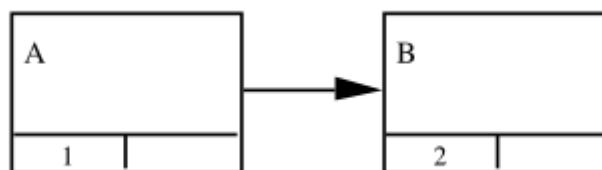


Рисунок 3.38 — Связь простой очередности между двумя UOB с именами A и B [39]

Реляционные связи (Relational Links) — связи обозначаемые прерывистыми линиями и не несущие никакой определённой семантики. Они могут указываться не только между различными UOB, но между сценариями и другими элементами диаграмм IDEF3. В целом они подчёркивают на существование (возможно, ограничивающее) между элементами диаграмм.

Связи приоритета ограничения (Constraint Precedence Links) — связи, включающие свойство простой очередности, но несущие дополнительную ограничивающую нагрузку между отношениями UOB. Такие связи имеют дополнительный графический элемент на стволе стрелки:

- символ треугольника* указывает своим углом направленность ограничения;
- символ ромба* указывает, что направленность ограничения распространяется на оба соединяемых UOB, в разных направлениях;
- символ прямоугольника* указывает, что свойства и направленность ограничения определены отдельным описанием или документом.

На рисунке 3.39 показаны две связи между UOB №1 с именем Sign timesheet (Подписать расписание) и UOB №2 с именем Obtain timesheet approval (Получить расписание):

- реляционная связь* указывает на некоторое «пользовательское» отношение;
- ограничивающая связь* указывает не только на временную очередность соединённых UOB, но и на то, что собственное расписание утверждать нельзя.

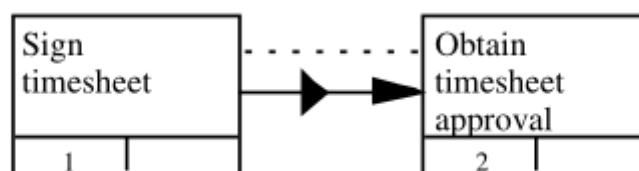


Рисунок 3.39 — Пример обозначения реляционной и ограничивающей связей между двумя UOB [39]

Методология IDEF3 поддерживает декомпозицию и собственную нумерацию блоков UOB в отображаемых диаграммах.

На рисунке 3.40 приведён абстрактный пример некоторой декомпозиции блоков UOB, одновременно поясняющий правила нумерации самих блоков UOB.

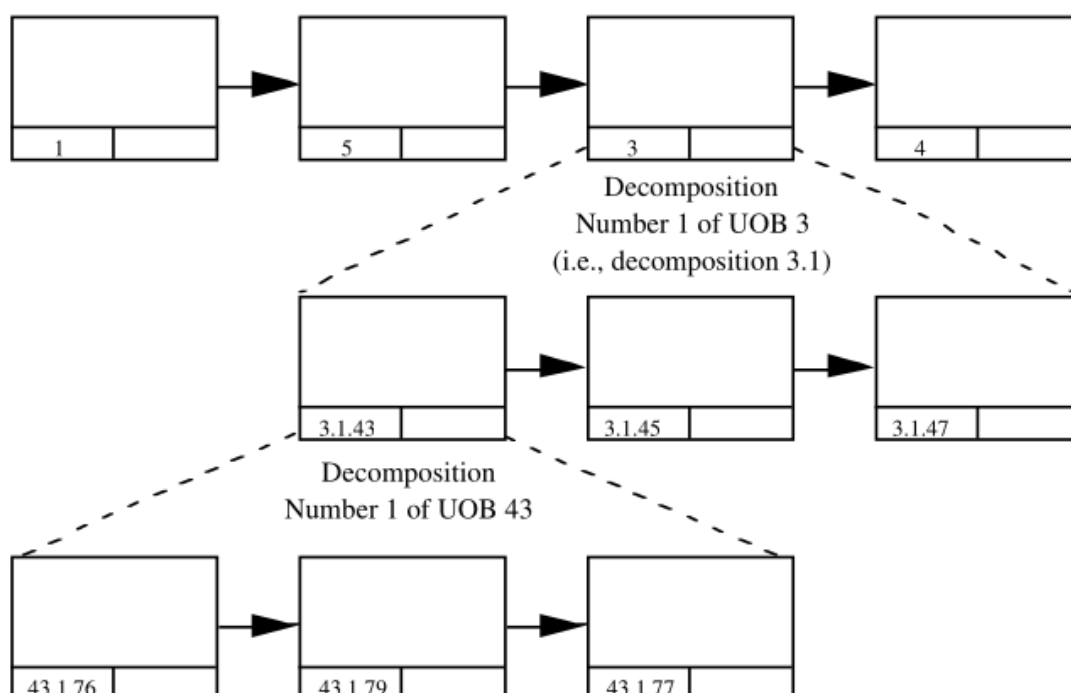


Рисунок 3.40 — Пример декомпозиции UOB в методологии IDEF3 [39]

Безусловно сам процесс декомпозиции естественным образом изменяет и уменьшает масштабность работ более нижнего уровня, но обратите внимание на нумерацию этих UOB, показанную в левой нижней их части. Хорошо видно, что блоки верхнего уровня нумеруются одним числом, а нижних уровней — тремя числами, разделёнными точками.

Правила нумерации блоков UOB:

- каждому разработчику диаграмм проекта выдаётся личный уникальный диапазон целых чисел, которые он может использовать для нумерации UOB верхнего уровня;
- если какой-либо UOB удаляется из диаграммы, то этот номер больше уже в проекте не используется;
- если UOB присутствует в различных вариантах, то номер этого варианта указывается после первой цифры, отделённой точкой;
- дочерние UOB, полученные в результате декомпозиции, нумеруются тремя целыми числами разделёнными точками: *Номер родительского UOB; Номер варианта родительского UOB; Номер UOB в пределах уровня декомпозиции.*




3.4.2 Работы и перекрёстки работ методологии IDEF3

Наиболее ценной особенностью методологии IDEF3 является возможность использования в диаграммах логических ограничений на последовательность и параллельность выполнения работ. Для реализации этой возможности используются *Перекрёстки* (Junctions).

Перекрёстки (Junctions) — геометрические элементы диаграмм методологии IDEF3, обеспечивающие отображение *слияний* (Fan-in) и *ветвлений* (Fan-out) работ с одновременным наложением на эти ситуации логических условий.

Синтаксис и семантика перекрёстков формально отображается информацией, представленной в таблице 3.4.

Таблица 3.4 — Синтаксис и семантика перекрёстков методологии IDEF3

Обозначение	Логический тип	Семантика слияния	Семантика ветвления
	Асинхронное AND	Все предшествующие процессы (работы) должны быть завершены	Все следующие процессы (работы) должны быть запущены
	Асинхронное OR	Один или несколько предшествующих процессов (работ) должны быть завершены	Один или несколько следующих процессов (работ) должны быть запущены
	Синхронное AND	Все предшествующие процессы (работы) завершены одновременно	Все следующие процессы (работы) запускаются одновременно
	Синхронное OR	Один или несколько предшествующих процессов (работы) завершаются одновременно	Один или несколько следующих процессов (работы) запускаются одновременно
	Эксклюзивное OR (XOR)	Только один предшествующий процесс (работа) завершён	Только один следующий процесс (работа) запускается

Все перекрёстки на диаграмме должны уникально нумероваться, причём каждый номер должен иметь префикс «**J**».

Безусловно наличие пяти логических условий создают большой потенциал для отображения большого количества вариантов построения диаграмм методологии IDEF3. Ниже, на рисунке 3.41, показан пример использования двух перекрёстков **J1** и **J2** для асинхронного согласования выполнения работ, обозначенных UOB с именами **A**, **B**, ..., **F** с помощью логической операции «**I**» (AND).

Согласно правилам представленным в таблице 3.4:

- после завершения работы **A**, запускаются работы **B**, **C** и **D**;
- работа **F** запускается только после того как завершатся работы **C**, **D** и **E**.

Естественным образом семантика указанных условий дополняется семантикой обозначений блоков UOB.

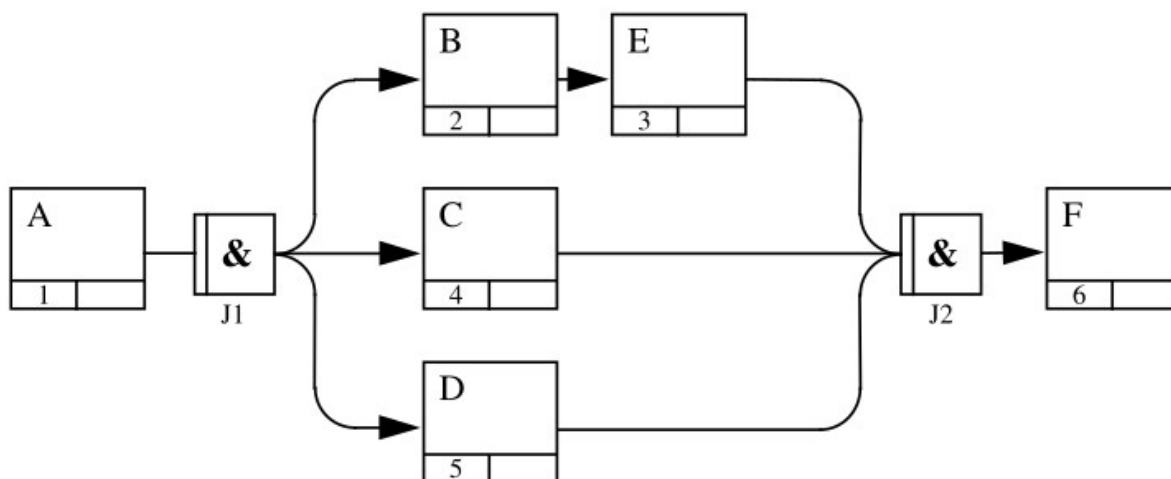


Рисунок 3.41 — Пример диаграммы IDEF3 с двумя перекрёстками [39]

3.4.3 Документирование моделей IDEF3

Кроме диаграмм, стандарт методологии IDEF3 [39] рекомендует оформление и использование множества различных текстовых документов, оформленных в виде специальных бланков. Один из примеров такого бланка показан на рисунке 3.42.

PROJECT LEADER:	DATE:	WORKING	REVIEWER:	DATE:
COMPANY:		DRAFT		
PROJECT NO.:	TASK NO.:	RECOMMENDED		
		RELEASED		
Purpose:				
Context:				
List of Scenarios:		List of Objects:		
DESCRIPTION NAME:			FORM TYPE:	
			Description Summary	

Рисунок 3.42 — Сводная форма описания IDEF3 [39]

Каждый рекомендуемый документ снабжён пояснениями и краткими инструкциями его использования. Во многом эти рекомендации соответствуют рекомендациям стандарта IDEF0.

3.4.4 Итоговая оценка применения методологии IDEF3

Методология структурного моделирования IDEF3 ориентирована на научное исследование, технологическое проектирование и разработку широкого класса автоматизированных систем, ориентированных на предметные области как бизнес-процессов, так и соответствующие области технологических процессов. Внешне указанные области кажутся достаточно близкими, для студентов — возможно одинаковыми. Но между ними имеются и существенные различия.

Бизнес-процесс — это логическая последовательность действий как отдельного человека, так и группы людей, в коллективе. Сама последовательность действий может осуществляться как на высшем уровне при принятии организационных решений, так и на уровне участия людей в технологических процессах производства. Но описательная и исследовательская сущность бизнес-процессов — регламентация самих этих действий в коллективе.

В плане методологии IDEF3 здесь конкурирует с теоретическими построениями идеологии Business Process Modeling (BPM).

Технологический процесс — последовательность действий (работ), которые осуществляются непосредственно с объектами производства, хотя и при участии людей.

В плане методологии IDEF3 здесь конкурирует с теоретическими построениями идеологии объектного подхода, в частности, — с идеей создания и использования унифицированного языка моделирования UML (*Unified Modeling Language*).

Таким образом, методология IDEF3 структурно стандартизирует как процессный, так и объектный подходы, описывая одновременно и методику их применения. Указанный факт уже был отмечен в начале данного подраздела посредством выделения двух методов: PFD и OSTD. Здесь же мы акцентируем внимание на указанном факте, чтобы подчеркнуть широкий охват предметных областей изученной методологией.

С другой стороны, излишняя универсальность методологии IDEF3 не делает её полностью идеальной:

- а) фактически требует построения двух видов диаграмм: процессных и объектных;
- б) является во многом избыточной, например, для проектирования информационных систем (ИС);
- в) слишком сложна в плане реализации инструментальных средств, полностью поддерживающих указанную методологию и автоматизирующих создание указанных инструментальных средств.

Вывод — Применительно к практике использования методологии IDEF3 для целей проектирования ИС следует ориентироваться на необходимость отображения последовательностей работ и особенно логических условий ограничения этих последовательностей, отображаемых инструментарием перекрёстков.

Что касается стадии концептуального проектирования ИС, то применение методологии IDEF3 следует осуществлять только в крайнем случае, когда методологии IDEF0 и DFD не обеспечивают нужного результата.

3.5 Моделирование потоков данных DFD

Последней методологией структурного функционального подхода проектирования ИС, заявленного в начале данного раздела, является моделирование потоков данных или сокращённо — *DFD*.

DFD (*Data Flow Diagrams*) — диаграммы потоков данных, составляющие графическую основу *нестандартизированной методологии* структурного функционального моделирования процессов обработки данных, ориентированных на проектирование информационных систем (ИС) различного назначения.

Методологию DFD можно и удобно рассматривать как прародительницу методологии IDEF0.

Сейчас считается, что первые упоминания о DFD относятся к середине 20-го века, а популярной эта методология стала в 1974 году, с момента выхода книги Эда Йордана и Ларри Константина «Структурное проектирование». Напомним, что набор методологий, обозначаемых как IDEF (ICAM Definition) появился только в 1981 году.

Популярность методологии DFD охватила умы различных исследователей информационных систем, благодаря своей простоте, опирающейся на четыре базовых понятия: *процесс*, *поток данных*, *внешняя сущность* и *хранилище данных*. Это позволило специалистам различных предметных областей и различной квалификации быстро находить общий язык.

Процесс (Process/Activity) — функция или последовательность действий (работ) моделируемой ИС, которую нужно выполнять, чтобы обработать поток входных данных и получить результат в виде потока выходных данных. Предполагается, что процесс, с целью анализа, может подвергаться декомпозиции.

Поток данных (Data flow) — информационный объект любого формата данных, который может быть обработан процессом, сгенерирован или потреблён внешней сущностью, а также сохранён в хранилище данных или извлечён из неё посредством процесса.

Внешняя сущность (External Entity/External Reference) — объект любой природы, находящийся за пределами моделируемой ИС и способный быть для процессов источником или потребителем потока данных.

Хранилище данных (Data store) — объект любой природы, находящийся внутри моделируемой ИС и способный сохранять или извлекать потоки данных для процессов. Хранилищами данных могут быть не только базы данных под управлением СУБД, но и файлы, бумаженные или иные носители информации. Если хранилище данных выносится за пределы моделируемой ИС, то оно превращается во внешнюю сущность.

К сожалению широкая популярность DFD привели к множеству интерпретаций и различным нотациям его графического языка. Тем не менее в нём сохранились черты, которые являются общими с чертами методологии IDEF0:

- а) требование наличия контекстной диаграммы, предполагающей графическое отображение единственного процесса, соответствующего функции самой ИС, необходимого окружения в виде внешних сущностей, определяющих внешнюю среду ИС в виде источников и потребителей информации, а также наличия «ЦЕЛИ» и «ТОЧКИ ЗРЕНИЯ», ограничивающих последующие толкования и интерпретации самой ИС;
- б) возможность декомпозиции процессов на необходимое число уровней;
- в) рекомендуемая градация уровней на позиции: *концептуального* или *контекстного* уровней; *логического* уровня и *физического* уровня;
- г) предполагается нумерация процессов;
- д) допускаются интерпретации результатов моделей как диаграмм «AS-IS» и «TO-BE».





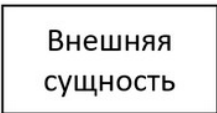
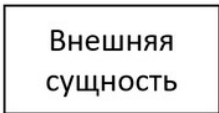
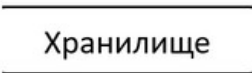

Чтобы конкретизировать теоретическое и практическое использование методологии DFD, рассмотрим учебный материал в следующих трёх аспектах:

- а) синтаксис и семантика языка DFD;
- б) учебный пример построения диаграмм DFD;
- в) выводы по применению методологии DFD и подведём итоги всего раздела.

3.5.1 Синтаксис и семантика языка DFD

Все графические элементы (компоненты) нотации методологии DFD должны быть именованы, дополнительно раскрывая семантику обозначаемых компонент. Исторически создано много нотаций этой методологии, а наиболее известными являются нотации Йордана-ДеМарко (Yordon-DeMarco) и Гейна-Сарсона (Gene-Sarson), показанные в таблице 3.5.

Таблица 3.5 — Два набора синтаксиса графических элементов методологии DFD

Компонента DFD	Нотация Йордана-ДеМарко	Нотация Гейна-Сарсона
Процесс		
Поток данных		
Внешняя сущность		
Хранилище данных		

Примечание — Имена элементов нотаций, представленных в таблице 3.5, являются метаопределениями имён соответствующих элементов, поэтому их нельзя использовать в диаграммах методологии DFD.

Дополнительно следует учесть множество нотаций, реализуемых различными средами разработки. В частности:

- а) с нотацией инструментальной среды BPWIN можно познакомиться в источнике [41];
- б) с нотацией инструментальной среды Ramus Educational для методологии IDEF0 можно познакомиться в источнике [42];
- в) нотация методологии DFD для среды Ramus Educational хорошо изложена в учебном пособии [43].

Учитывая, что студент уже хорошо познакомился с общими правилами построения функциональных диаграмм на примерах учебного материала технологий IDEF0 и IDEF3, дальнейшее описание применения синтаксиса и семантики методологии DFD проведём на основе конкретного учебного примера, изложенного в следующем пункте.

3.5.2 Постановка учебной задачи по методологии DFD

Ранее, для целей концептуального проектирования ИС ЭЖР, студент Петров И.В. провёл функциональное моделирование системы с помощью методологии IDEF0. Диаграммы этого моделирования представлены в разделе 3.3 данного пособия в виде рисунков 3.24 — 3.27.

Анализ диаграммы *A2* (рисунок 3.26), которая продублирована ниже в виде рисунка 3.43, показал, что заказчику не понятно: каким образом может быть реализован блок *A23* «Ответить на вопрос студента».

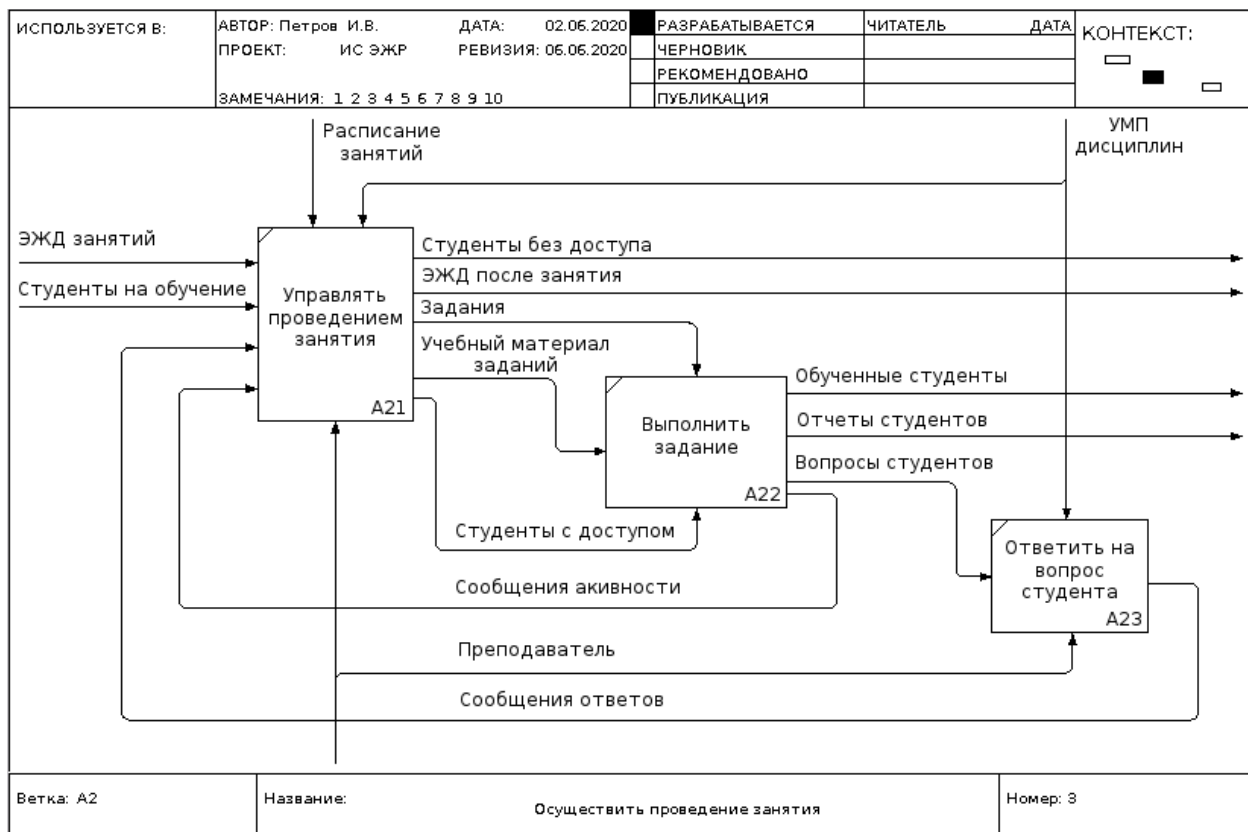


Рисунок 3.43 — Дубль рисунка 3.26, отображающего диаграмму A2 ИС ЭЖР

Объяснения Петрова И.В. о том, что блок A23 может быть реализован с использованием любой СУБД не убедил заказчика («Потенциального заказчика»), который потребовал привести дополнительные декомпозиции, более наглядно поясняющие возникший вопрос.

Поскольку диаграммы IDEF0 не отображают хранилища данных, то было *принято решение*: пояснить структуру функционирования блока A23 посредством методологии DFD.

Рассмотрим блок A23 как отдельную маленькую информационную систему с именем «Ответить на вопрос студента».

Согласно диаграмме A2, блок A23:

- а) *принимает потоки данных* «Вопросы студентов», поступающие из блока A22, и поток данных в виде текстов ответов преподавателя, которые не показаны явно на рисунке 3.43, но подразумеваются в виде связи механизма «Преподаватель»;
- б) *генерирует поток данных* «Сообщения ответов», который поступает в функциональный блок A21.

Следует также обратить внимание на два дополнительных обстоятельства:

- а) поток данных «Вопросы студентов» генерируется механизмом «Студенты с доступом», который в методологии DFD можно рассматривать как внешнюю сущность «Студенты с доступом»;
- б) элементы потоков данных «Вопросы студентов» и «Сообщения ответов» имеют внутреннюю связь, которая преобразуется в элементы данных «Вопрос-ответ» и могут рассматриваться как элементы хранилища данных «БД вопросов и ответов».

Проведённого краткого анализа диаграммы **A2** уже достаточно для формулирования учебной задачи, нацеленной на реализацию принятого ранее решения.

Учебная задача — выполнить построение контекстной диаграммы и необходимую её декомпозицию для блока **A23**, в соответствии с правилами методологии DFD.

Решение этой учебной задачи начнём с формализации компонент блока **A23** для уровня контекстной диаграммы и изложим необходимые операции и действия в отдельных пунктах данного подраздела.

3.5.3 Формализация компонент контекстной диаграммы блока **A23**

Этап формализации компонент любой методологии не только значительно облегчает последующее построение диаграмм, но и является частью проектной документации любой системы.

Поскольку методология DFD — формально не стандартизирована, то имеет смысл использовать рекомендации, например, методологии IDEF0.

Мы же воспользуемся результатами рассуждений предыдущего пункта, а итоговый перечень компонент представим в виде таблицы 3.6.

Таблица 3.6 — Набора компонент методологии DFD для учебной задачи

Компонент DFD	Семантика компонента
Процесс «Ответить на вопрос студента»	Функциональный блок контекстной диаграммы, соответствующий функциональному блоку A23
Хранилище данных «БД вопросов и ответов»	Внутренний элемент ИС, семантически соответствующий некоторой таблице базы данных
Внешняя сущность «Студенты с доступом»	Источник генерации потока данных «Вопросы студентов»
Внешняя сущность «Преподаватель»	Источник генерации потока данных «Тексты ответов» и потребитель потока данных «Тексты вопросов»
Поток данных «Вопросы студентов»	Связь: «Студенты с доступом» - Процесс
Поток данных «Сообщения ответов»	Связь: Процесс - Хранилище данных
Поток данных «Тексты вопросов»	Связь: Процесс - «Преподаватель»
Поток данных «Тексты ответов»	Связь: «Преподаватель» - Процесс
Поток данных «Вопросы на регистрацию»	Связь: Процесс - Хранилище данных
Поток данных «Вопросы для ответа»	Связь: Хранилище данных - Процесс

Примечание — Обратите внимание, что в методологии DFD не допускаются прямые связи между внешними сущностями и базами данных. Другими словами — хранилища данных и внешние сущности имеют связи только с процессами.

Завершение процесса формализации компонент контекстной диаграммы должно завершаться формулировкой цели и точки зрения создаваемого структурного проекта.

ЦЕЛЬ: Структурное описание блока A23 в нотации DFD.

ТОЧКА ЗРЕНИЯ: Заказчик проекта ИС ЭЖР.

Используя проведённую формализацию компонентов DFD учебной задачи, перейдём к этапу реализации самой контекстной диаграммы.

3.5.4 Реализация контекстной диаграммы в системе Ramus Educational

Учитывая, что студент Петров И.В. использует инструментальное средство Ramus Educational, то у него имеются два варианта решения поставленной выше задачи:

- а) *первый вариант* — взять диаграмму блока A2, показанную выше на рисунке 3.43, выделить на ней блок A23 и создать декомпозицию этого блока, указав, что используется нотация DFD;
- б) *второй вариант* предполагает открытие нового проекта с созданием контекстной диаграммы в нотации DFD и проведение всех необходимых этапов её декомпозиции.

Примечание — Мы приведём пример использования второго варианта, одновременно демонстрируя все особенности реализации проекта в нотации DFD.

Запустив инструментальную среду системы Ramus, выберем режим «Создание нового проекта» и, в мастере «Свойства проекта», укажем тип нотации DFD, как это показано на рисунке 3.44.

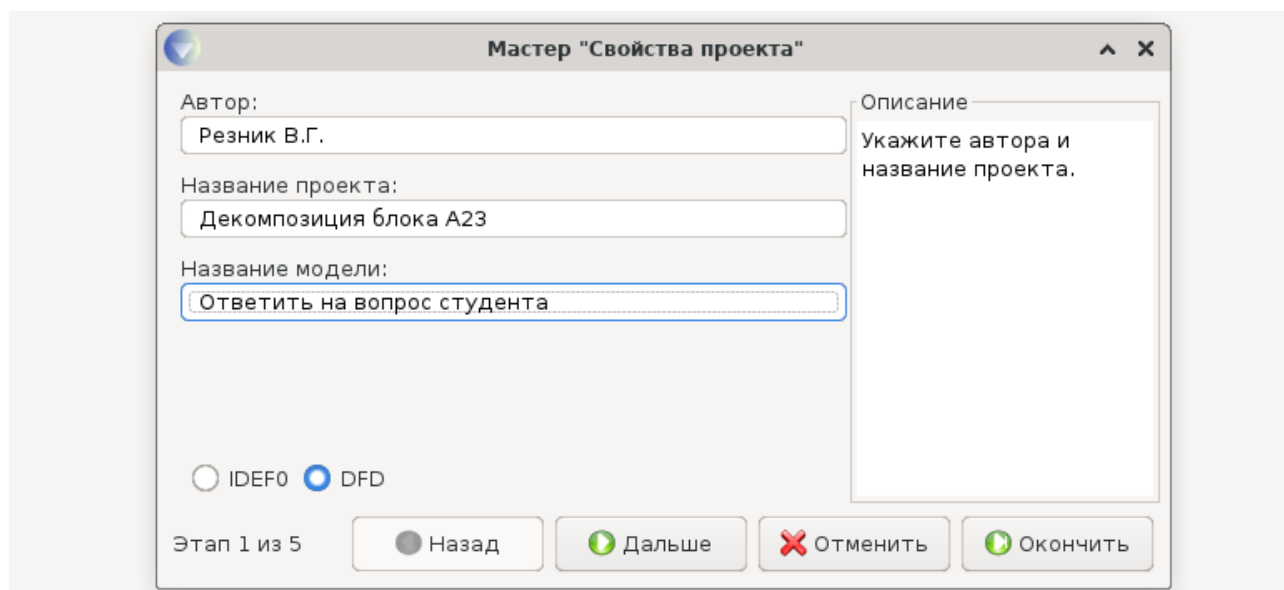


Рисунок 3.44 — Создание нового проекта в нотации DFD

Нажав кнопку «Дальше», переходим к этапу 2 и указываем, где используется проект, например, как показано на рисунке 3.45.

Нажимая кнопку «Дальше», перейдём к этапу 4, где в мастере «Свойства проекта» заполним список классификаторов, указав в них имена хранилища данных и внешних сущностей из таблицы 3.6. Результат этого этапа отображён на рисунке 3.46.

Классификатор — это набор уникальных названий объектов, являющихся особенностью реализации системы Ramus и используемых этот набор для различных целей. Имена классификаторов присваиваются *хранилищам* и *внешним сущностям*.

Для целей проектирования ИС — классификаторы можно рассматривать как *таблицы будущих баз данных*. Для этого имена классификаторов рассматриваются как названия соответствующих таблиц, а *атрибуты классификаторов* могут рассматриваться как имена и типы столбцов, указанных таблиц.

Атрибуты — единый дополнительный список объектов системы Ramus, каждый из которых создаётся как пара: «Имя атрибута» : «Тип атрибута». В дальнейшем атрибуты могут распределяться (связываться) с классификаторами.

Примечание — При создании нового проекта необходимо создать хотя бы один классификатор, иначе файл проекта может создаваться с ошибками.

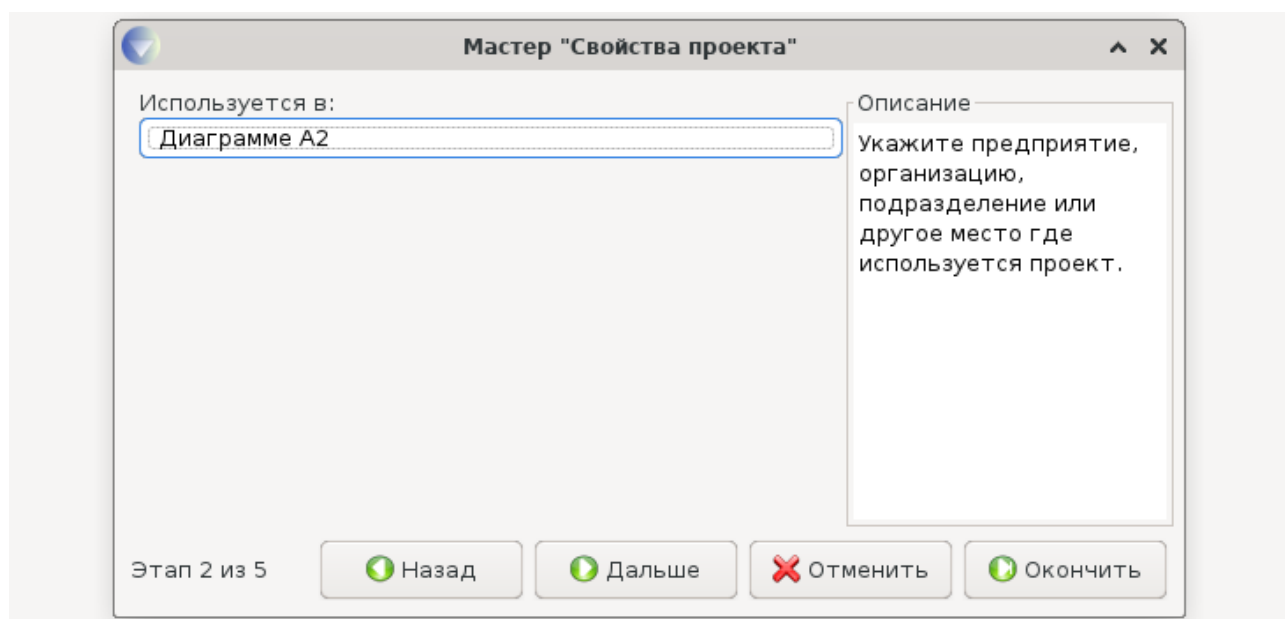


Рисунок 3.45 — Второй шаг создания проекта

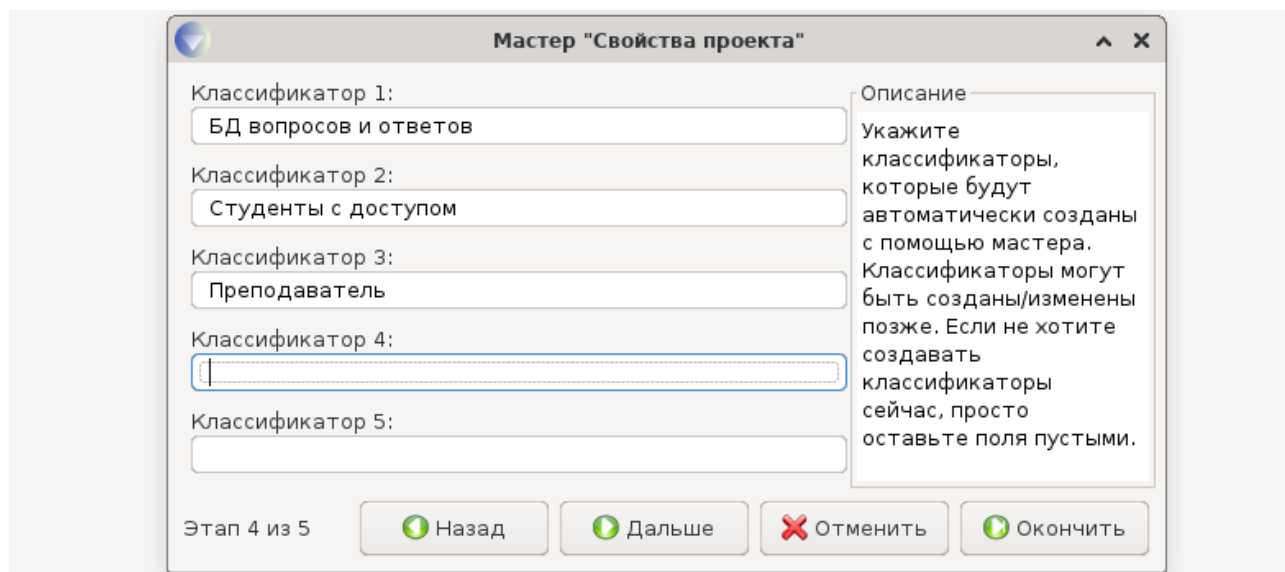


Рисунок 3.46 — Создание классификаторов для хранилища данных и внешних сущностей

После записи классификаторов можно нажать кнопку «Окончить» и система Ramus откроет пустое окно проекта.

Обязательно нужно сохранить созданный проект, нажав комбинацию клавиш **Ctrl-S**. В результате появится окно, показанное на рисунке 3.47. После нажатия клавиши «Сохранить», созданный проект будет сохранён в файле «Пример блока A23.rsф».

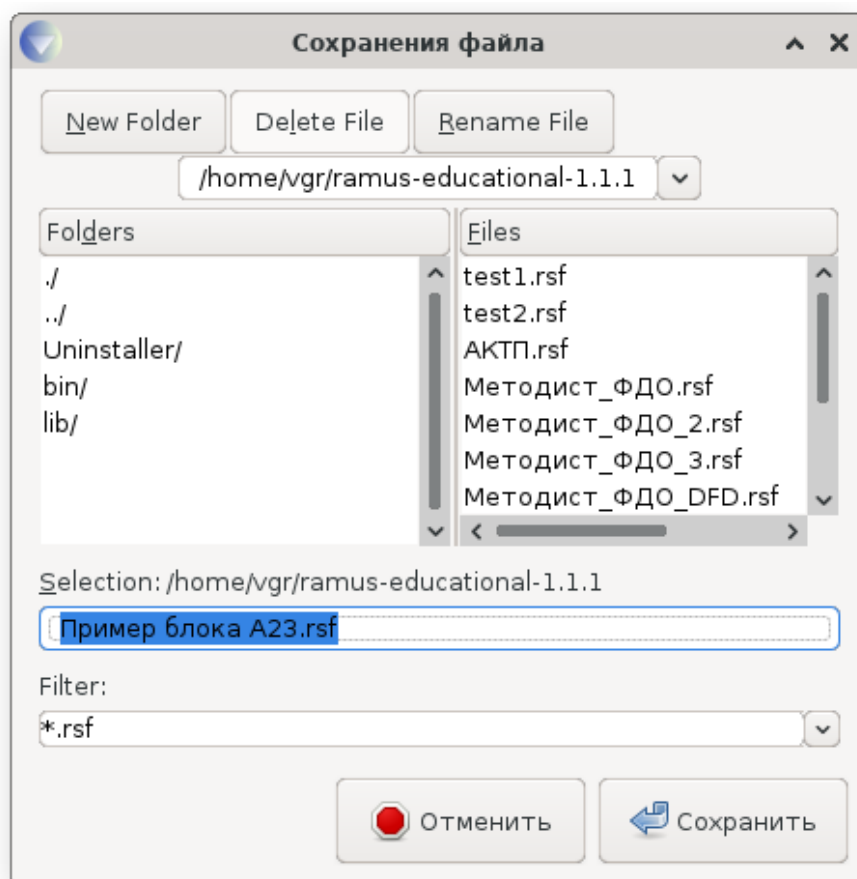


Рисунок 3.47 — Окно сохранения проекта в системе Ramus

Сохранив созданный проект, приступаем к размещению на диаграмме компонент DFD согласно информации размещённой в таблице 3.6.

Сначала размещаем компоненты процесса, хранилища данных и внешних сущностей.

Затем хранилище данных и внешние сущности привязываем к уже созданным классификаторам, для чего:

- выделяем курсором мыши нужный компонент и активируем контекстное меню правой кнопкой мыши;
- в появившемся контекстном меню выбираем пункт «Редактировать активный элемент» и в появившемся окне «Свойства DFD объекта» нажимаем кнопку «Задать DFD объект»;
- в новом окне «Выберите классификатор» выделяете мышкой нужный классификатор и закрываете окна, нажимая на клавиши «ОК».

В результате указанных действий, все хранилища данных и внешние сущности будут связаны с соответствующими классификаторами и получают их целочисленные имена.

Процесс создание контекстной диаграммы Завершается добавлением компонент потоков данных, цели и точки зрения. Результат построения диаграммы показан на рисунке 3.48.

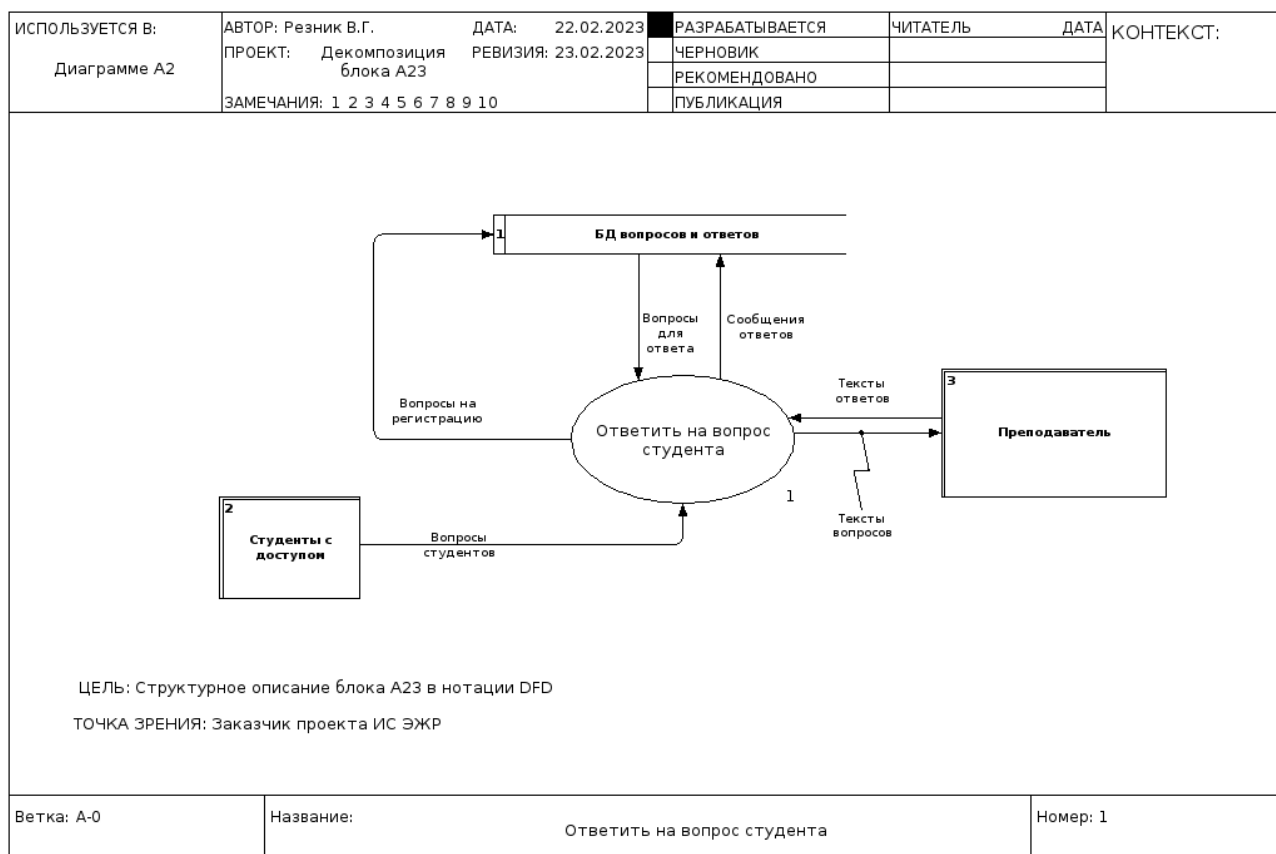


Рисунок 3.48 — Контекстная диаграмма блока А23 в нотации DFD

После согласования с заказчиком было принято решение провести декомпозицию контекстной диаграммы учебной задачи.

3.5.5 Декомпозиция контекстной диаграммы учебной задачи

В результате анализа контекстной диаграммы, студент Петров И.В. решил выделить три дочерних блока. Результат декомпозиции представлен в таблице 3.7.

Таблица 3.7 — Блоки декомпозиции процесса «Ответить на вопрос студента»

№ п/п	Имя блока	Семантика блока
1	Сохранить вопрос студента	Данный блок принимает входной поток данных «Вопросы студентов» и преобразует его в выходной поток данных «Вопросы на регистрацию», обеспечивая сохранение вопросов в хранилище данных. Такой подход обеспечивает <i>асинхронность функционирования внешних сущностей</i> «Студенты с доступом» и «Преподаватель».
2	Выбрать вопрос	Блок читает поток «Вопросы для ответа» и передаёт его во внешнюю сущность «Преподаватель».
3	Записать сообщение ответа	Блок принимает поток данных «Тексты ответов» и формирует выходной поток «Сообщения ответов», направляя его в хранилище данных «БД вопросов и ответов».

На основании информации этой таблицы, студент Петров И.В. приступил к реализации диаграммы декомпозиции учебной задачи.

Результат декомпозиции представлен на рисунке 3.49.

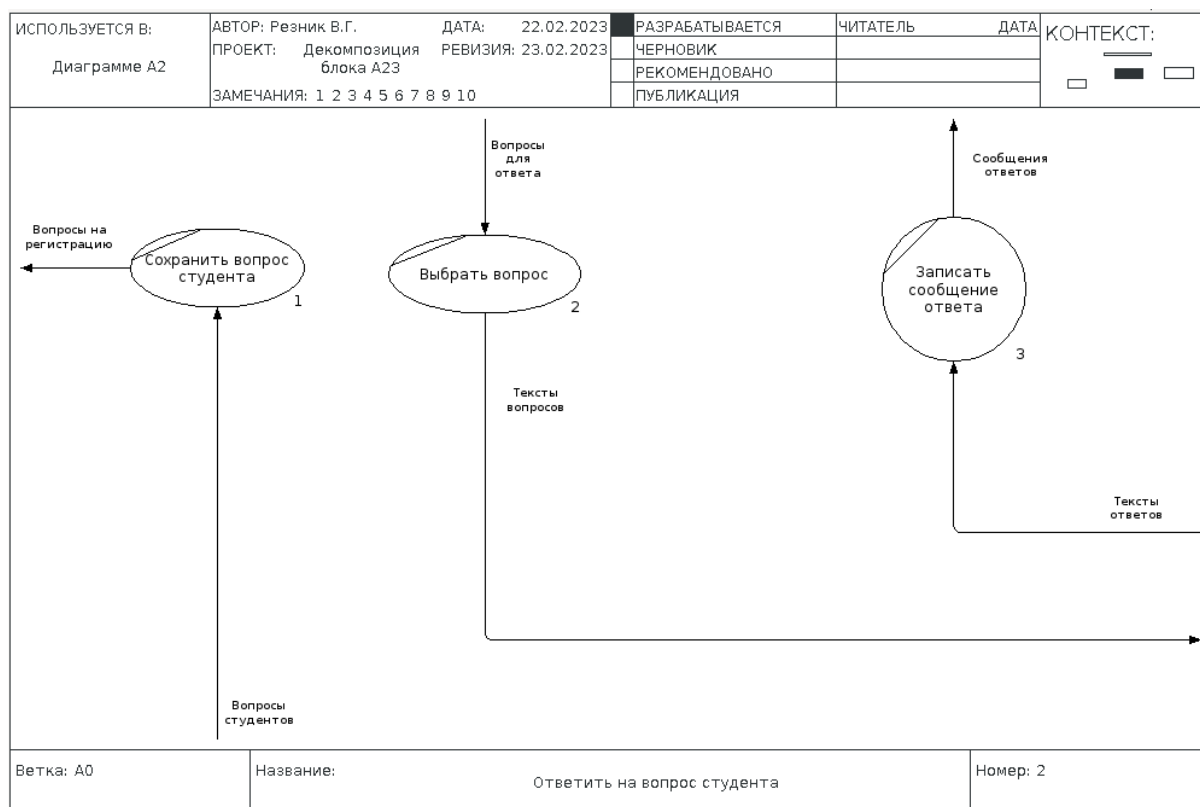


Рисунок 3.49 — Декомпозиция контекстной диаграммы с выделенными тремя процессами

Приведённая декомпозиция наглядно показывает, что:

- а) процесс генерации вопросов студентами и процесс генерации ответов преподавателем осуществляются асинхронно, не задерживая и не ожидая работы друг друга; это в полной мере поясняет функционирование блока А23;
- б) объекты хранилищ данных и внешних функций вынесены за поле диаграммы декомпозиции, в результате чего стрелки потоков данных направлены в разные стороны; это требует напряжение воображения читателя для восстановления всей детальной картины связей между объектами блока А23.

В целом графическое представление рисунка 3.49 во многом даже проигрывает графическому представлению контекстной диаграммы, показанной на рисунке 3.48.

Выявленный негативный результат реализации двух диаграмм, выполненных по методологии DFD, является общим негативным свойством всех структурных диаграмм с иерархическим типом зависимостей.

После согласования с заказчиком было принято решение о реализации одноуровневого представления диаграммы учебной задачи.

3.5.6 Итоговая одноуровневая диаграмма блока А23 в нотации DFD

При создании нового проекта система Ramus требует создание контекстной диаграммы А-0, поэтому она была реализована без внешних связей, как это показано на рисунке 3.50. Результат декомпозиции, с учётом наличия хранилища данных и внешних сущностей, представлен на рисунке 3.51. Полученный результат удовлетворил требования заказчика.

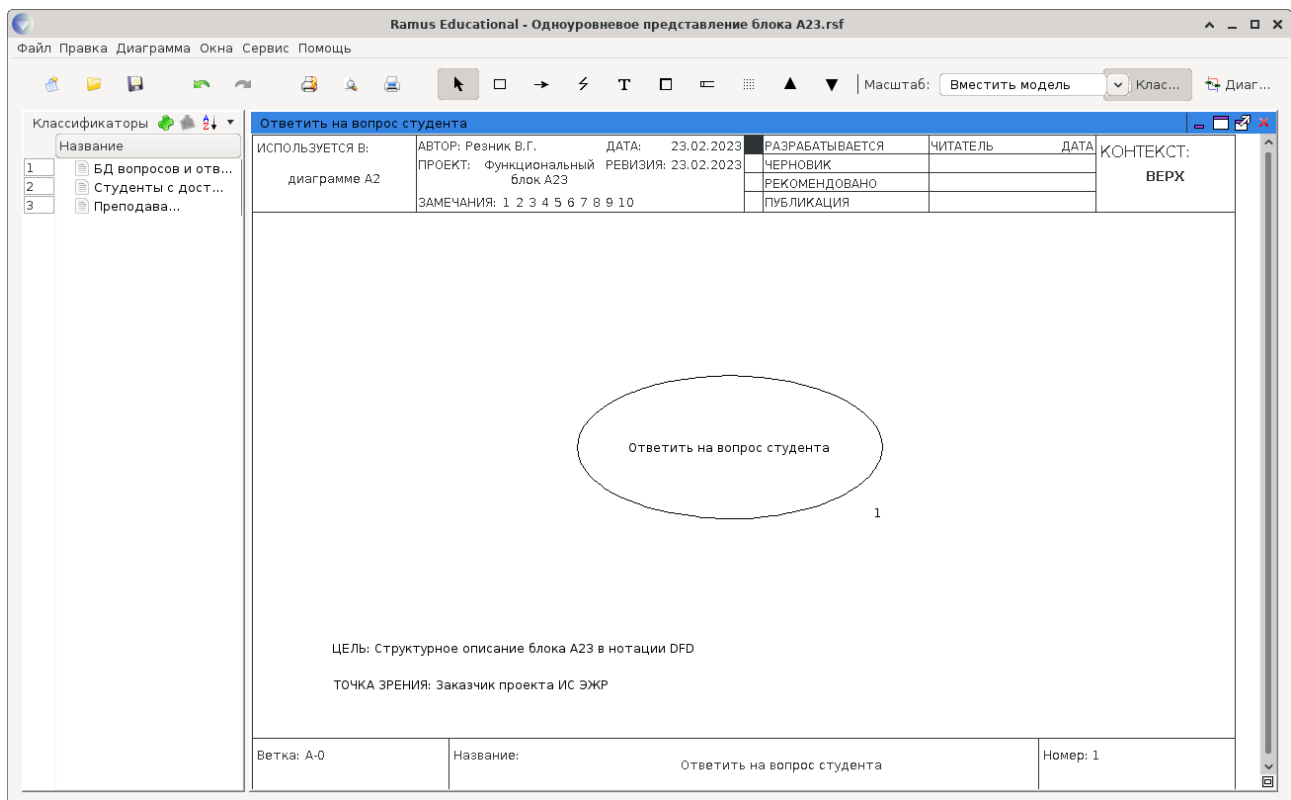


Рисунок 3.50 — Второй вариант контекстной диаграммы учебной задачи в нотации DFD

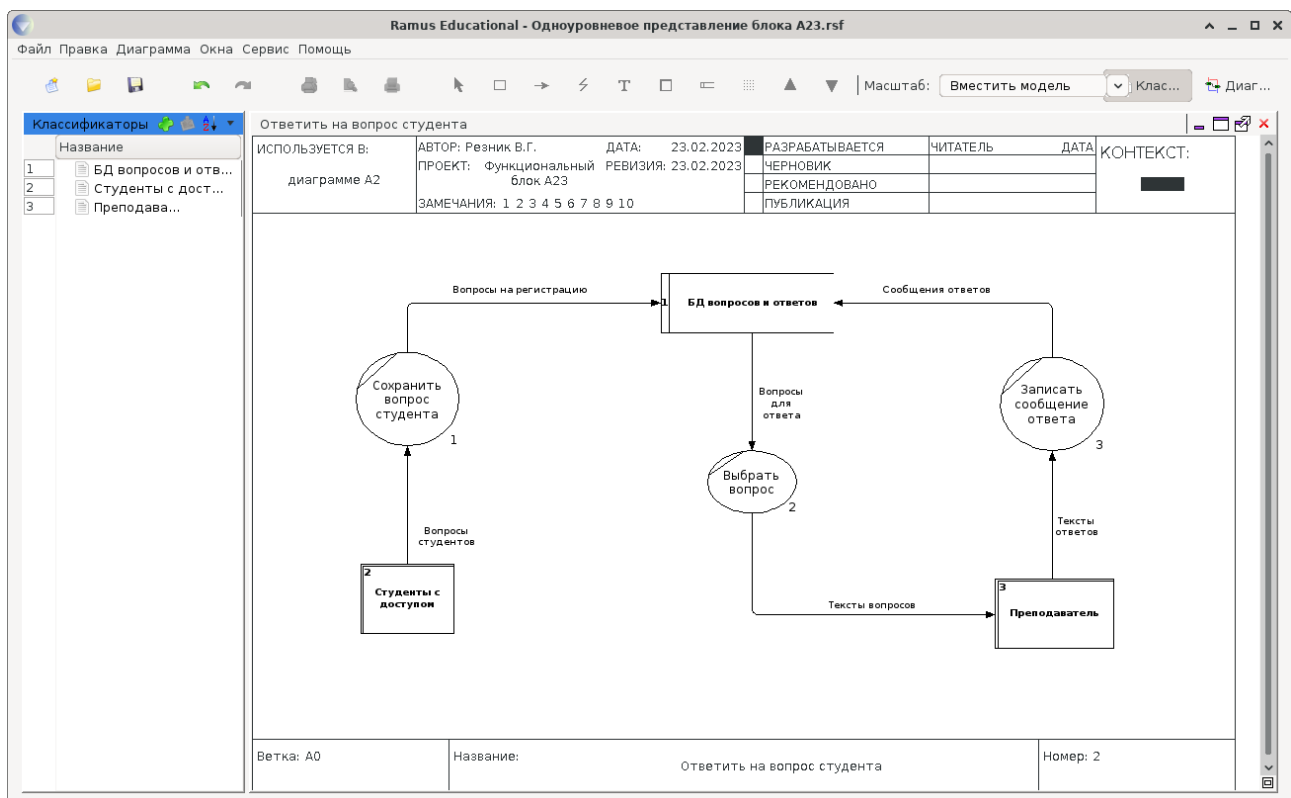


Рисунок 3.51 — Второй вариант декомпозиции учебной задачи в нотации DFD

На этом примере мы заканчиваем изучение методологии DFD и переходим к обсуждению итогов обучения по материалу всего раздела.

3.6 Выводы по применению методологий концептуального проектирования ИС

Стадия концептуального проектирования ИС предназначена для формирования ТТЗ (Тактико-Технического Задания) или ТКП (Технико-Коммерческого Предложения) с целью убедить будущего Заказчика ИС сформировать ТЗ (Техническое Задание) и заключить договор на проектирование и реализацию ИС.

Будущий (потенциальный) Заказчик воспринимает ТТЗ или ТКП как набор функций, совокупность которых должна удовлетворить его бизнес-коммерческие потребности. Именно поэтому основу стадии концептуального проектирования составляет деятельность по функциональному моделированию ИС. Более того, поскольку ИС является компонентой АС (Автоматизированной Системы), то будущий (потенциальный) Исполнитель должен использовать адекватную нотацию (формальный язык), который понятен как самому себе, так и потенциальному заказчику. Такие адекватные нотации присущи методологиям структурного моделирования: IDEF0, IDEF3 и DFD.

В общем случае, все методологии структурного моделирования связаны с описанием и декомпозицией процессов, но по разному используют свои графические нотации и интерпретацию процессов как функций.

Методология IDEF0 рассматривает функцию как *преобразователь потоков материальных и информационных объектов*, указывая в явном виде ограничения на функцию, в виде управления, и явно задавая активные субъекты (механизмы), обеспечивающие как само преобразование объектов, так и учёт ограничений. При таком потенциале IDEF0 позволяет потенциальному Исполнителю концептуально проектировать не только ИС, но и — АС. Именно поэтому методология IDEF0 является обязательной на рассмотренной здесь стадии.

Методология IDEF3 рассматривает функцию как *работу*, а процессы как *последовательности, возможно параллельные, таких работ*. На уровне концептуального проектирования как ИС, так и — АС, такая нотация может не потребоваться, поэтому данная методология рассматривается как дополнительная.

Методология DFD рассматривает функцию как *процесс, преобразующий входные потоки данных в выходные потоки данных*. В такой метасемантике эта методология казалось бы должна идеально подходить для целей концептуального проектирования ИС, но её практическое использование упирается в ряд проблем:

- а) формальное отсутствие стандартизации и наличие множества графических нотаций;
- б) отсутствие ограничивающих условий управления;
- в) отсутствие явного выделения механизмов исполнения;
- г) второстепенное значение структуры и форматов обрабатываемых потоков данных.

Все эти недостатки методологии DFD делают её второстепенной по сравнению с методологией IDEF0 и необходимой только для пояснения структурных деталей средствами, которые отсутствуют в самой методологии IDEF0.

В целом, большое многообразие проектируемых ИС требует, чтобы студент уверенно владел каждой из перечисленных методологий и максимально эффективно использовал их на стадии концептуального проектирования ИС и на стадии формирования технического задания (ТЗ).

Вопросы для самопроверки

1. Что обозначают сокращения SADT и ICAM, а также в чём состоит различие обозначаемых ими понятий?
2. Каково назначение и чем заканчивается стадия концептуального проектирования?
3. Что такое — функциональное моделирование бизнес-процессов и какими средствами оно осуществляется?
4. Чем отличаются методологии IDEF0, IDEF3 и DFD и на каких стадиях проектирования ИС они применяются?
5. Перечислите стандартный состав проектной группы, рекомендуемый методологией IDEF0?
6. Что такое — контекстная диаграмма методологии IDEF0 и что на ней должно быть отображено?
7. Что такое — ICOM-кодирование и какие правила существуют для адресации блоков на диаграммах методологии IDEF0?
8. Что такое — туннелирование на диаграммах IDEF0 и как оно обозначается?
9. Перечислите обязательный набор документов для проекта IDEF0?
10. Какие типы диаграмм позволяет создавать методология IDEF3?
11. Что означает сокращение UOB применительно к методологии IDEF3?
12. Перечислите элементы графического языка описания процессов, предоставляемые методологией IDEF3?
13. Какие информационные элементы содержатся в блоке UOB?
14. Какие перекрёстки работ методологии IDEF3 вам известны?
15. Что такое — методология DFD и чем она отличается от методологий IDEF0 и IDEF3?
16. Назовите базовые элементы графической нотации методологии DFD?
17. Какие типы нотаций методологии DFD вам известны?
18. Что понимается под процессом в методологии DFD?
19. Что такое в методологии DFD — внешняя сущность и чем она отличается от потока данных?
20. Чем в методологии DFD внешняя сущность отличается от хранилища данных?

ПЗ ЛАБОРАТОРНАЯ РАБОТА №3

Концептуальное проектирование ИС индивидуальной учебной задачи

ПЗ.1 Общая постановка задачи

Выполнить стадию «2 Концептуальное проектирование ИС» применительно к индивидуальной учебной задаче, используя учебный материал описанного выше подраздела 3 данного методического пособия и фактографический материал лабораторной работы №2.

ПЗ.2 Содержательная часть лабораторной работы

Выполнить концептуальное проектирование учебной задачи, используя методологию IDEF0 и инструментальное средство Ramus Educational.

В отдельных пунктах отчёта по лабораторной работе изложить:

- 1) **Контекстную диаграмму** индивидуальной учебной задачи.
- 2) **Диаграммы** двух уровней декомпозиции контекстной диаграммы.
- 3) **Глоссарий**, содержащий в алфавитном порядке список сокращений или наборов слов, использованных в именах блоков и стрелок, а также раскрывающий их точное смысловое содержание.
- 4) **Текст**, раскрывающий и комментирующий в текстовом виде графические изображения диаграмм.

ПЗ.3 Список использованных источников

1. Резник В.Г. Проектирование информационных систем. Методические указания по самостоятельной и индивидуальной работе студента / В.Г. Резник. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2023. – 17 с. [Электронный ресурс]: — Режим доступа: pis_self_20_09.03.01.pdf.
2. ГОСТ 34.601-90 — АС. АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ, 1990. – 9 с. [Электронный ресурс]: — Режим доступа: ГОСТ 34.601-90 — АС. Стадии создания.pdf.
3. РС Р 50.1.028-2001 — МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ. — М.: ИПК Издательство стандартов, 2001. — 54 с. [Электронный ресурс]: — Режим доступа: РС Р 50.1.028-2001-IDEF0.pdf.