Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра "Вычислительная техника"

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ № 2

DFD-технология: интегрированная структурная модель

по дисциплине

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Направление подготовки: 09.03.03 Прикладная информатика
Профиль подготовки: Прикладная информатика в промышленности
Квалификация выпускника: бакалавр

Форма обучения: заочная

Тула 2021

Методические указания составлены Сычуговым А.А. и обсуждены на заседании кафедры *«Информационная безопасность»* Института прикладной математики и компьютерных наук.

Зав. кафедрой _____ А.А. Сычугов

Содержание

Содержание 3
1. Цель и задачи работы 4
2. Общие положения (теоретические сведения) 4
3. Объекты исследования, оборудование, материалы и наглядные пособия 15
4. Задание на работу (рабочее задание) 15
5. Ход работы (порядок выполнения работы) 15
6. Содержание отчета 15
7. Список использованных источников 16

1. Цель и задачи работы

Цель работы – освоить навыки построения диаграммы потоков данных для сложных систем.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- Освоить принципы построения модели DFD;
- Представить диаграмму потоков данных для сложной системы с помощью нотации DFD по индивидуальному заданию.

2. Общие положения (теоретические сведения)

В основе классической DFD-технологии лежат три группы средств моделирования:

• диаграммы, иллюстрирующие функции, которые система должна выполнять, и связи между этими функциями - для этой цели используются собственно диаграммы потоков данных DFD (Data Flow Diagrams), дополненные словарями данных и спецификациями процессов нижнего уровня;

• диаграммы, моделирующие данные и их взаимосвязи, - для этой цели используются диаграммы «сущность-связь» ERD (Entity-Relationship Diagrams);

• диаграммы, моделирующие поведение системы, - для этой цели используются диаграммы переходов состояний STD (State Transition Diagrams).

Все эти диаграммы содержат графические и текстовые средства моделирования: первые - для удобства демонстрирования основных компонентов модели, вторые - для обеспечения точного определения ее компонентов и связей.

Основа модели - диаграмма DFD показывает внешние по отношению к системе адресаты и адресанты данных, идентифицирует логические функции (процессы) и группы элементов данных, связывающие одну функцию с другой (потоки), а также идентифицирует накопители (хранилища) данных, к которым осуществляется доступ. Структуры потоков данных и определения их компонентов хранятся и анализируются в словаре данных. Каждая логическая функция (процесс) может быть детализирована с помощью DFD ниж-

него уровня; когда дальнейшая детализация перестает быть полезной, переходят к выражению логики функции при помощи спецификации процесса нижнего уровня (мини-спецификации). Содержимое каждого накопителя также сохраняют в словаре данных, модель данных накопителя раскрывается с

помощью ERD. В случае построения поведенческой модели DFD дополняется средствами описания, зависящего от времени поведения системы, раскрывающегося с помощью STD. Эти связи показаны на рис. 1.

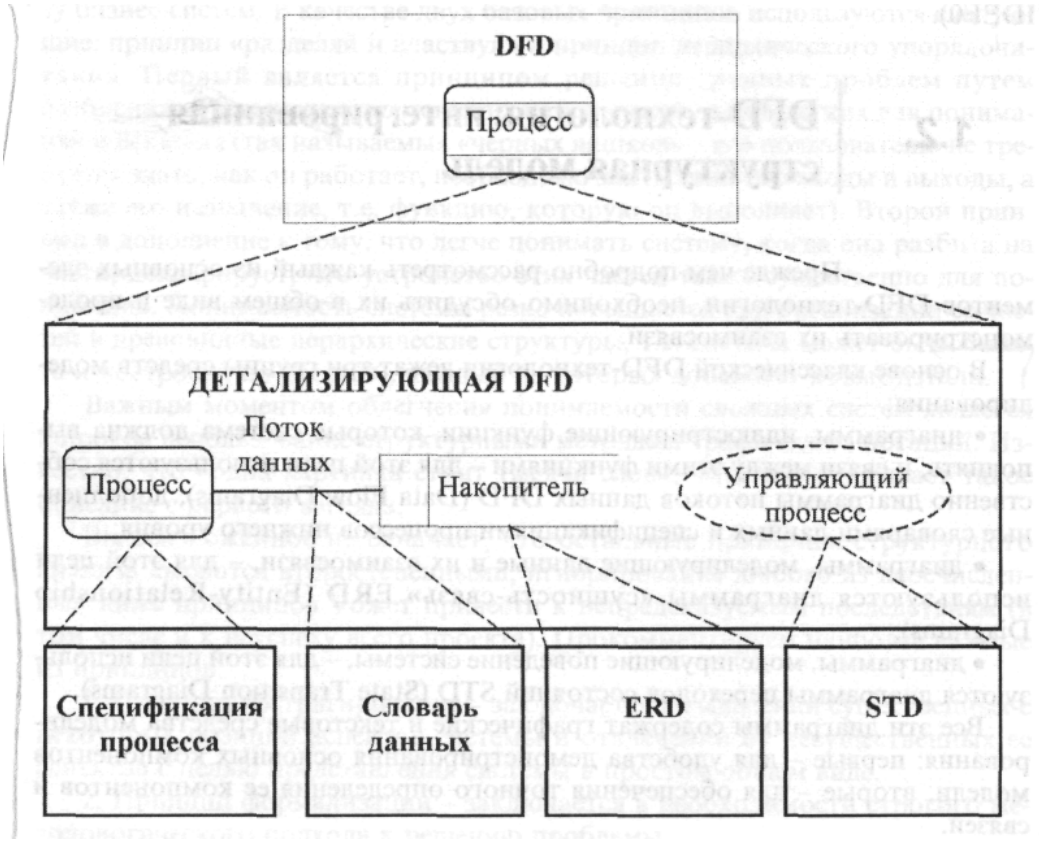


Рисунок 1 - Компоненты DFD-технологии

1 Иерархическая функциональная модель

Диаграммы потоков данных (DFD) являются основным средством функционального моделирования бизнес-системы. С их помощью система разбивается на функциональные компоненты (процессы) и представляется в виде сети, связанной потоками данных. Главная цель таких средств - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Диаграммы потоков данных известны очень давно. В фольклоре

упоминается следующий пример использования DFD для реорганизации переполненного клерками офиса, относящийся к 1920-м годам. Осуществлявший реорганизацию консультант обозначил кружком каждого клерка, а стрелкой -каждый документ, передаваемый между ними. Используя такую диаграмму, он предложил схему реорганизации, в соответствии с которой двое клерков, интенсивно обменивающихся большим количеством документов, были посажены рядом, а клерки с малым взаимодействием были посажены на большом расстоянии. Так родилась первая модель, представляющая собой потоковую диаграмму, - предшественник DFD.

Диаграммы потоков данных чрезвычайно просты, наглядны и понятны. Базовая их нотация включает всего четыре рассматриваемых ниже объекта.

1. Поток данных является механизмом, использующимся для моделирования передачи информации (или даже физических компонентов) из одной части системы в другую. Важность этого объекта очевидна: он дает название целому инструменту. Потоки на диаграммах обычно изображаются именованными стрелками (при этом имя потока отражает его содержимое), ориентация которых указывает направление движения информации.

2. Назначение процесса состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Это имя должно содержать глагол в неопределенной форме с последующим дополнением (например, ВЫЧИСЛИТЬ МАКСИМАЛЬНУЮ ВЫСОТУ) или отглагольное существительное (ВЫЧИСЛЕНИЕ МАКСИМАЛЬНОЙ ВЫСОТЫ). Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели. Процессы обозначаются на диаграмме, как показано на рис. 1.2; символ процесса включает три разделенных горизонтальными чертами поля:

• верхнее поле содержит номер процесса и аббревиатуру детализирующего объекта (КД - контекстная диаграмма, ДПД - диаграмма потоков дан-

ных, МС - мини-спецификация);

• среднее поле содержит имя процесса;

• нижнее поле содержит имя исполнителя процесса (идентификатор подразделения, должности, механизма и т.п.).

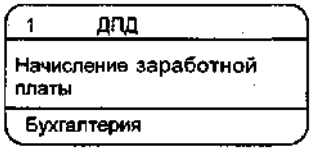


Рисунок 2 - Изображение процесса на диаграммах

3. Накопитель данных позволяет на определенных участках определять данные, которые будут сохраняться вне процессов. Фактически накопитель представляет «срезы» потоков данных во времени. Информация, которую он содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя накопителя должно идентифицировать его содержимое и быть существительным во множественном числе.

Логически накопители данных есть некие устройства для хранения информации, куда ее можно поместить и через некоторое время извлечь. При этом не уточняется способ помещения и извлечения данных в накопитель, нас не интересует, происходит ли извлечение данных для чтения (копирования) или для изъятия и другие подобные вопросы.

Накопитель данных обозначается, как показано на рис. 1.3. Каждый накопитель данных идентифицируется для ссылки буквами «БД» и числом в квадрате с левой стороны, определяющим его номер. Имеется возможность создавать копии накопителя, в этом случае номер соответствующей копии фиксируется под идентификатором накопителя.

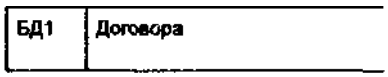


Рисунок 3 - Изображение накопителя на диаграммах

Когда процесс сохраняет данные, то стрелка потока данных направлена в накопитель данных, и, наоборот, когда доступ в накопитель данных осу-

ществляется для чтения, стрелка потока данных направлена в процесс.

4. Внешняя сущность представляет сущность вне контекста системы, являющуюся источником или приемником системных данных, например ЗАКАЗЧИК, ПОСТАВЩИК, СКЛАД ТОВАРОВ. Определение некоторого объекта в качестве внешней сущности указывает на то, что он находится за пределами анализируемой системы. Предполагается, что такие объекты не должны участвовать ни в какой обработке. Внешняя сущность обозначается на диаграмме, как показано на рис. 1.4.



Рисунок 4 - Изображение внешней сущности на диаграммах

В качестве примера DFD рассмотрим верхний уровень функциональной модели компании, занимающейся распределением товаров по заказам (рис. 1.5). Заказы подвергаются входному контролю и сортировке. Если заказ не отвечает номенклатуре товаров или оформлен неправильно, то он аннулируется с соответствующим уведомлением заказчика. Если заказ не аннулирован, то определяется, имеется ли на складе соответствующий товар. В случае положительного ответа выписывается счет к оплате и предъявляется заказчику, при поступлении платежа товар отправляется заказчику. Если заказ не обеспечен складскими запасами, то отправляется заявка на товар производителю. После поступления требуемого товара на склад компании заказ становится обеспеченным и повторяет вышеописанный маршрут.

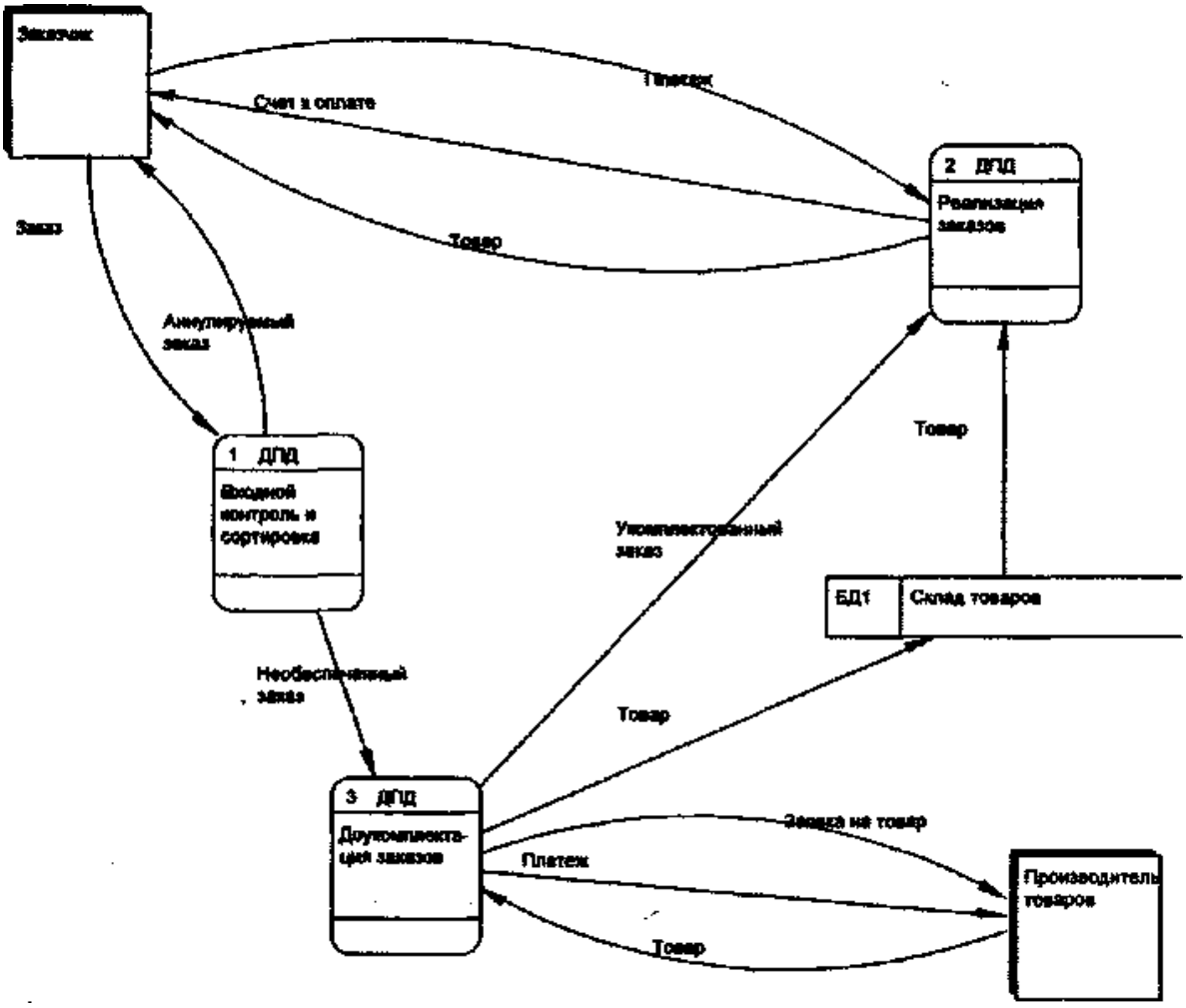


Рисунок 5 - Пример диаграммы потоков данных

При детализации процессов часто возникает необходимость в детализации (структуризации) потоков данных. Например, на диаграмме верхнего уровня может находиться один информационный поток, а при детализации на следующих уровнях иерархии может выясниться, что данный поток есть результат слияния нескольких потоков. Такое слияние осуществляется через информационный канал. Логически информационный канал есть среда передачи информации. Он может реализоваться в виде, например, курьерской службы, почты, магистрали или шины данных и т.п.

Обозначающий канал символ имеет поле имени и поле номера копии, как показано на рис. 6.

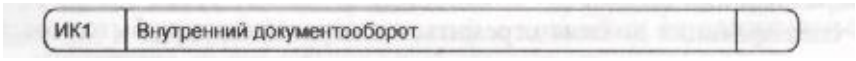


Рисунок 6 - Изображение информационного канала на диаграммах

Как уже отмечалось, декомпозиция DFD осуществляется на основе

процессов: каждый процесс может раскрываться с помощью DFD нижнего уровня.

Моделирование поведения системы

Управляющий поток представляет собой «трубопровод», через который проходит управляющая информация. Его имя не должно содержать глаголов, а только существительные и прилагательные. Обычно управляющий поток имеет дискретное, а не непрерывное значение. Это может быть, например, сигнал, представляющий состояние или вид операции.

Управляющий процесс представляет собой интерфейс между ПРО и спецификациями управления, собственно моделирующими и документирующими аспекты поведения системы. Его имя указывает на тип управляющей деятельности. Фактически управляющий процесс представляет собой преобразователь входных управляющих потоков в выходные управляющие потоки; при этом точное описание этого преобразования должно задаваться в спецификации управления. Логически управляющий процесс есть некий командный пункт, реагирующий на изменения внешних условий, передаваемые ему с помощью управляющих потоков, и продуцирующий в соответствии со своей внутренней логикой выполняемые процессами команды.

Спецификации управления предназначены для моделирования и документирования поведенческих аспектов систем, они позволяют осуществлять декомпозицию управляющих процессов и описывают отношения между входными и выходными управляющими потоками на детализируемом управляющем процессе.

Для этой цели в DFD-технологиях обычно используются диаграммы переходов состояний (STD), содержащие следующие объекты.

Состояние - объект, рассматривающийся как условие устойчивости для системы. Находясь в определенном состоянии, система имеет достаточно информации о ее прошлой истории, чтобы определить очередное состояние в

зависимости от текущих входных событий. Особую роль играет начальное состояние - узел STD, являющийся стартовой точкой для начала функционирования системы. STD имеет ровно одно начальное состояние, соответствующее состоянию системы после ее инсталляции, но перед началом реальной обработки, а.также любое (конечное) число завершающих состояний.

Переход - объект, определяющий перемещение моделируемой системы из одного состояния в другое. При этом имя перехода идентифицирует условие, являющееся причиной перехода и управляющее им. Это условие обычно продуцирует управляющий поток (сигнал), возникающий как во внешнем мире,и так и внутри моделируемой системы. Кроме условия с переходом может связываться ***действие*** или ряд действий, выполняющихся, когда переход имеет место.

Фактически условие есть некоторое внешнее или внутреннее событие, которое система способна обнаружить и на которое она должна отреагировать определенным образом, изменяя свое состояние. При изменении состояния система обычно выполняет одно или более действий, представляющих собой отклик, посылаемый во внешнее окружение, или вычисление, результаты которого запоминаются в системе (обычно в накопителях данных на DFD) для того, чтобы обеспечить реакцию на некоторые из планируемых в будущем событий.

На STD состояния представляются узлами, а переходы - дугами (рис. 1.7). Условия идентифицируются именем перехода и возбуждают выполнение перехода. Действия привязываются к переходам и записываются под соответствующим условием. Начальное состояние на диаграмме должно иметь входной переход, изображаемый потоком из подразумеваемого стартового узла (иногда этот стартовый узел изображается небольшим квадратом и привязывается к входному состоянию).



Рисунок 7 - Символы STD

Информационная модель системы

DFD-технологии предполагают использование диаграмм «сущность-связь» (ERD) для разработки информационных моделей и обеспечения стандартного способа определения данных и отношений между ними. Фактически с помощью ERD осуществляется детализация накопителей данных DFD-диаграммы, а также документируются информационные аспекты бизнес-системы, включая идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений).

Сущность представляет собой множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, идей, предметов и т.п.), обладающих общими атрибутами или характеристиками. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. При этом имя сущности должно отражать тип или класс объекта, а не его конкретный экземпляр (например, АЭРОПОРТ, а не ВНУКОВО).

Отношение в самом общем виде представляет собой связь между

двумя и более сущностями. Фактически отношение является ассоциацией между сущностями, при которой каждый экземпляр одной сущности ассоциируется с произвольным (в том числе и нулевым) количеством экземпляров другой сущности, и наоборот. Именование отношения осуществляется с помощью грамматического оборота глагола (ИМЕЕТ, ОПРЕДЕЛЯЕТ, МОЖЕТ ВЛАДЕТЬ и т.п.).

Другими словами, сущности представляют собой базовые типы информации, необходимые системе, а отношения показывают, как эти типы данных взаимоувязаны друг с другом.

Для идентификации требований, в соответствии с которыми сущности вовлекаются в отношения, используются связи. Каждая связь соединяет сущность и отношение и может быть направлена только от отношения к сущности. Значение связи характеризует ее тип и, как правило, выбирается из следующего множества: {«О или 1», «О или более», «1», «1 или более», «p:q» (диапазон)}. Пара значений связей, принадлежащих одному и тому же отношению, определяет тип этого отношения. Практика показала, что для моделирования большинства систем достаточно использовать следующие ,типы отношений:

1) 1*1 (одип-к-одному). Отношения данного типа используются, как правило, на верхних уровнях иерархии модели данных, а на нижних уровнях встречаются сравнительно редко;

2) 1*n (один-ко-многим). Отношения данного типа являются наиболее часто используемыми;

3) n*n (многие-ко-многим). Отношения данного типа обычно используются на ранних этапах моделирования с целью прояснения ситуации. В дальнейшем каждое из таких отношений должно быть преобразовано в комбинацию отношений типов 1 и 2 (возможно, с добавлением вспомогательных сущностей и с введением новых отношений).

Сущность на ERD представляется прямоугольником любого размера, содержащим внутри себя имя сущности, список имен атрибутов (возможно,

(неполный) и указатели ключевых атрибутов (знак «#» перед именем атрибута).

Отношения представляются линиями, соединяющими сущности и состоящими из двух частей (соответствующих паре связей), для которых должны быть определены имя, степень множественности (один или много объектов участвуют в связи) и степень обязательности (т.е. обязательная или необязательная связь между сущностями). Для множественной связи линия присоединяется к прямоугольнику сущности в трех точках, а для одиночной связи - в одной точке. При обязательной связи рисуется непрерывная линия до середины отношения, при необязательной - пунктирная линия. Вышесказанное иллюстрируется на рис. 1.8.

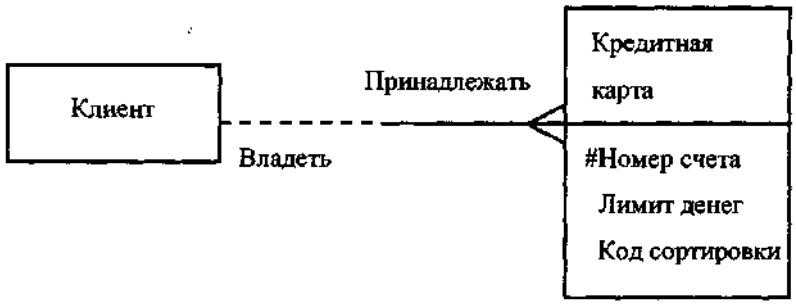


Рисунок 8 - Символы ERD

Читается отношение отдельно по каждой связи, демонстрируя, как сущность КЛИЕНТ связывается с сущностью КРЕДИТНАЯ КАРТА, и наоборот. При этом необходимо учитывать степень обязательности выбранного конца связи, для этой цели используются слова «должен (быть)» или «может (быть)». Так, диаграмма, приведенная на рис. 1.7, читается следующим образом:

Каждый КЛИЕНТ может ВЛАДЕТЬ одной или более КРЕДИТНОЙ КАРТОЙ

или

Каждая КРЕДИТНАЯ КАРТА должна ПРИНАДЛЕЖАТЬ ровно одному КЛИЕНТУ.

3. Объекты исследования, оборудование, материалы и наглядные пособия

Объектом исследования данной лабораторной работы являются технологические процессы при разработке сложной системы.

Для выполнения работы необходимы ПК и соответствующее программное обеспечение:

- MS WINDOWS;
- MS Visio
- MS Office (для оформления отчета).

4. Задание на работу (рабочее задание)

Исследовать структуру, связи и поведение сложной системы с помощью DFD - методологии.

Разработать технологический процесс для следующих объектов:

1. Отчет по л/р
2. Отчет по КР
3. ПО
4. Комплекс АСУ
5. Изделие машиностроительного производства
6. Пирог
7. Стул

5. Ход работы (порядок выполнения работы)

Для выполнения лабораторной работы необходимо:

1) Ознакомиться с теоретической справкой;

2) Выявить основные сущности сложной системы как процесса;

3) Выявить взаимосвязи между сущностями, их тип и атрибуты;

4) Построить диаграмму потоков данных DFD процесса, дополненные словарями данных и спецификациями процессов нижнего уровня;

5) Построить диаграмму «сущность-связь» ERD;

6) Построить диаграмму переходов состояний STD;

7) Оформить отчет по лабораторной работе;

8) Защитить работу преподавателю, ответив на вопросы по ее содержанию и выполнению.

6. Содержание отчета

В отчете должны присутствовать следующие пункты:

1) задание;

2) диаграмма потоков данных DFD процесса, словари и спецификации;

3) диаграмма «сущность-связь» ERD;

4) диаграмма переходов состояний STD.

7. Список использованных источников

Ямникова О.А. Разработка САПР: учеб. пособие / О.А. Ямникова. – Тула: Издв-во ТулГУ, 2010 – 67 с.