

**Диаграммы потоков данных (DFD - Data Flow Diagrams)** являются основным средством моделирования функциональных требований к проектируемой системе. С их помощью эти требования представляются в виде иерархии функциональных компонентов (процессов), связанных потоками данных. Главная цель такого представления - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Для построения DFD традиционно используются две различные нотации, соответствующие методам Йордана и Гейна-Сарсона. Эти нотации незначительно отличаются друг от друга графическим изображением символов. Далее при построении примеров будет использоваться нотация Гейна-Сарсона.

Для изображения диаграмм потоков данных традиционно используют два вида нотаций: нотации Йордана и Гейна-Сарсона (рисунок 1).


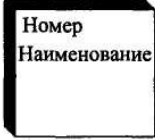


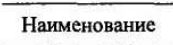

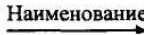

Понятие	Нотация Йордана	Нотация Гейна-Сарсона
Внешняя сущность		
Система, подсистема или процесс		
Накопитель данных		
Поток		

Рисунок 1

Модель системы в контексте DFD представляется в виде некоторой информационной модели, основными компонентами которой являются различные потоки данных, которые переносят информацию от одной подсистемы к другой. Каждая из подсистем выполняет определенные преобразования входного потока данных и передает результаты обработки информации в виде потоков данных для других подсистем.

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те, в свою очередь, преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям — потребителям информации.

Таким образом, основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы/подсистемы;
- процессы;
- накопители данных;
- потоки данных.

#### Состав диаграмм потоков данных (процессов)

**Внешняя сущность** представляет собой материальный объект или физическое лицо, представляющие собой источник или приемник информации, например заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что они находятся за пределами границ анализируемой системы. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь

диаграммы анализируемой системы, если это необходимо, или, наоборот, часть процессов может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Внешняя сущность обозначается квадратом (рис. 2), расположенным как бы над диаграммой и бросающим на нее тень для того, чтобы можно было выделить этот символ среди других обозначений.

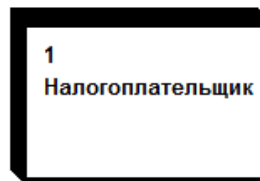


Рисунок 2 - Графическое изображение внешней сущности

При построении модели сложной ЭИС она может быть представлена в самом общем виде на так называемой *контекстной диаграмме* в виде одной **системы** как единого целого либо может быть декомпозирована на ряд **подсистем** (рис. 3).

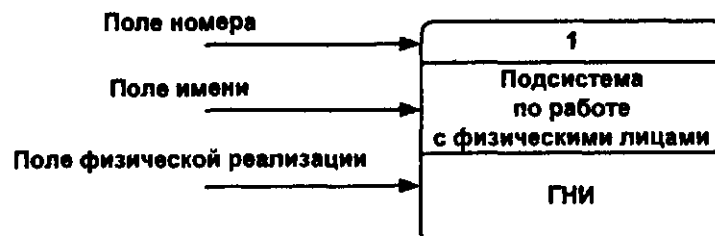


Рисунок 3 - Подсистема по работе с физическими лицами (ГНИ — Государственная налоговая инспекция)

Номер подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

**Процесс** представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т. д. Процесс на диаграмме потоков данных изображен на рис. 4.

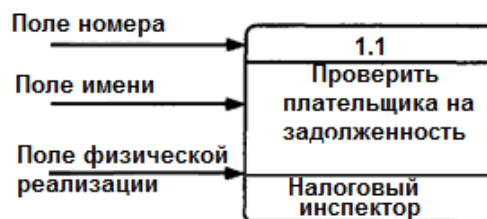


Рисунок 4 - Графическое изображение процесса

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

"Ввести сведения о налогоплательщиках",  
"Выдать информацию о текущих расходах",  
"Проверить поступление денег".

Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать", означает недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

**Накопитель данных** - это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т. д. Накопитель данных на диаграмме потоков данных (рис. 5) идентифицируется буквой "D" и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных, и описание хранящихся в нем данных должно быть увязано с информационной моделью (ERD).

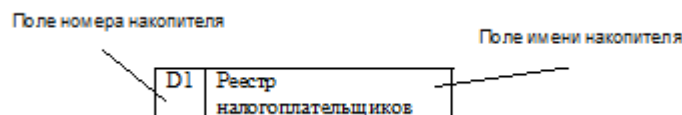


Рисунок 5 - Графическое изображение накопителя данных

**Поток данных** определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой, и т. д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока (рис. 6). Каждый поток данных имеет имя, отражающее его содержание.

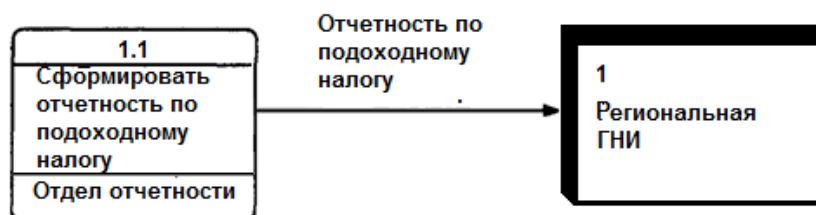


Рисунок 6 - Поток данных

### Построение иерархии диаграмм потоков данных

Построение иерархии диаграмм потоков данных начинают с диаграммы особого вида — **контекстной диаграммы**, которая определяет наиболее общий вид системы. На такой диаграмме показывают, как разрабатываемая система будет взаимодействовать с приемниками и источниками информации без указания исполнителей, т.е. описывают интерфейс между системой и внешним миром.

Если проектируемая система содержит большое количество внешних сущностей (более 10-ти), имеет распределенную природу или включает уже существующие подсистемы, то строят *иерархии* контекстных диаграмм.

При разработке контекстных диаграмм происходит детализация функциональной структуры будущей системы, что особенно важно, если разработка ведется несколькими коллективами разработчиков.

На следующем этапе каждую подсистему контекстной диаграммы детализируют при помощи диаграмм потоков данных. В процессе детализации соблюдают правило балансировки - **при детализации подсистемы можно использовать компоненты только тех подсистем, с которыми у разрабатываемой подсистемы существует информационная связь** (т. е. с которыми она связана потоками данных).

Для облегчения восприятия процессы детализируемой подсистемы нумеруют, соблюдая иерархию номеров: так процессы, полученные при детализации процесса или

подсистемы «1», должны нумероваться «1.1», «1.2» и т. д. Кроме этого, желательно размещать на каждой диаграмме от 3-х до 6-7-ми процессов и не загромождать диаграммы деталями, не существенными на данном уровне.

Декомпозицию потоков данных необходимо осуществлять параллельно с декомпозицией процессов.

Полная спецификация процессов включает также описание *структур данных*, используемых как при передаче информации в потоке, так и при хранении в накопителе. Описываемые структуры данных могут содержать *альтернативы*, *условные вхождения* и *итерации*.

**Условное вхождение** означает, что соответствующие элементы данных в структуре могут отсутствовать.

**Альтернатива** означает, что в структуру может входить один из перечисленных элементов. **Итерация** означает, что элемент может повторяться некоторое количество раз.

Кроме того, для данных должен быть указан тип: непрерывное или дискретное значение. Для непрерывных данных могут определяться единицы измерений, диапазон значений, точность представления и форма физического кодирования. Для дискретных - может указываться таблица допустимых значений.

Полученную законченную модель необходимо проверить на полноту и согласованность. Под **согласованностью** модели в данном случае понимают выполнение для всех потоков данных *правила сохранения информации*: все поступающие куда-либо данные должны быть считаны и записаны.

В качестве примера рассмотрим упрощенную модель процесса получения некоторой суммы наличными по кредитной карточке клиентом банка. Внешними сущностями данного примера являются клиент банка и, возможно, служащий банка, который контролирует процесс обслуживания клиентов. Накопителем данных может быть база данных о состоянии счетов отдельных клиентов банка. Отдельные потоки данных отражают характер передаваемой информации, необходимой для обслуживания клиента банка. Соответствующая модель для данного примера может быть представлена в виде диаграммы потоков данных (рисунок 7).

В настоящее время диаграммы потоков данных используются в некоторых CASE-средствах для построения информационных моделей систем обработки данных. Основным недостаток этой методологии также связан с отсутствием явных средств для объектно-ориентированного представления моделей сложных систем, а также для представления сложных алгоритмов обработки данных. Поскольку на диаграммах DFD не указываются характеристики времени выполнения отдельных процессов и передачи данных между процессами, то модели систем, реализующих синхронную обработку данных, не могут быть адекватно представлены в нотации DFD.



Рисунок 7 - Пример диаграммы DFD для процесса получения некоторой суммы наличными по кредитной карточке

**Пример 1.** Разработаем иерархию диаграмм потоков данных программы построения графиков/таблиц функций.

Разработку начнем с построения контекстной диаграммы. Для чего определим внешние сущности и потоки данных между программой и внешними сущностями. У данной системы единственная внешняя сущность Учащийся. Он вводит или выбирает из списка функцию, задает интервал и количество точек, а затем получает таблицу значений функции и ее график. На рисунке 8 представлена контекстная диаграмма системы.

Детализируя эту диаграмму, получаем три процесса: *Ввод/выбор функции и ее разбор*, *Построение таблицы значений функции* и *Построение графика функции*. Для хранения функций добавляем хранилище функций. Затем определяем потоки данных.

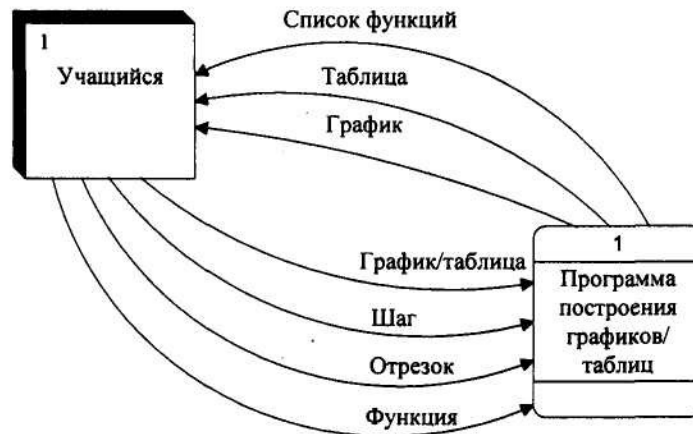


Рисунок 8 - Контекстная диаграмма программы построения графиков функций (нотация Гейна-Сарсона)

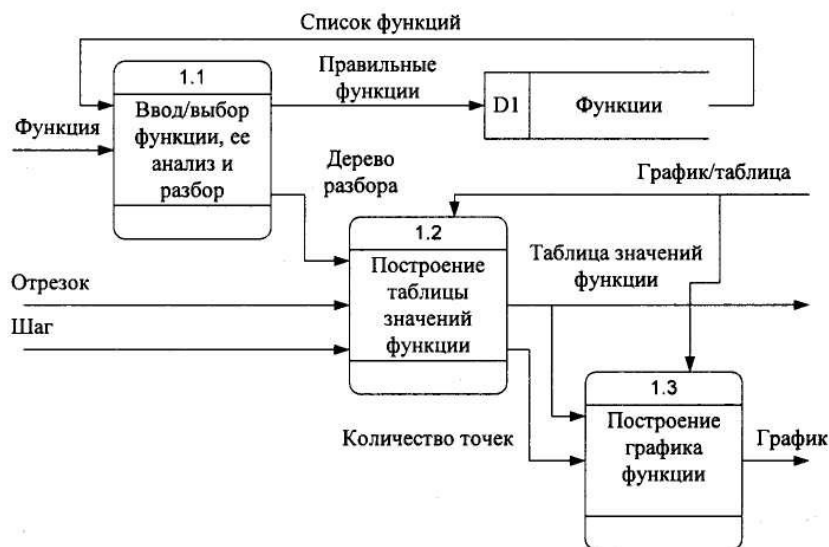


Рисунок 9 - Детализирующая диаграмма потоков данных системы исследования функций (нотация Гейна-Сэрсона)

Например, на диаграмме потоков данных можно показать хранилище данных, что очень существенно для систем, включающих базы данных. Кроме того, диаграммы потоков данных позволяют точно адресовать функции системы при наличии нескольких категорий пользователей, что демонстрирует следующий пример.

**Пример 2.** Разработать иерархию диаграмм потоков данных системы учета успеваемости студентов.

В качестве внешних сущностей для системы выступают Декан, Заместитель декана по курсу и Сотрудник деканата. Определим потоки данных между этими сущностями и системой.

Декан должен получать (рисунок 10):

- сводку успеваемости по факультету (процент успеваемости групп, курсов и в целом

по факультету) на текущий или указанный момент времени;

- полные сведения об учебе конкретного студента (успеваемость по всем изученным предметам всех завершенных семестров обучения с учетом пересдач).

Заместитель декана по курсу должен получать:

- сводку успеваемости по курсу (процент успеваемости по группам) на текущий или указанный момент;
- сведения о сдаче экзаменов и зачетов указанной группой;
- текущие сведения об успеваемости конкретного студента;
- полные сведения об учебе конкретного студента (успеваемость по всем изученным предметам всех завершенных семестров обучения с учетом пересдач);
- список задолжников по факультету с указанием групп и несданных предметов.

Сотрудник деканата должен обеспечивать:

- ввод списков студентов, зачисленных на первый курс;
- корректировку списков студентов в соответствии с приказами о зачислении, отчислении, переводе и т.п.;
- ввод учебных планов кафедр;
- ввод расписания сессии;
- ввод результатов сдачи зачетов и экзаменов на основании ведомостей и направлений.

Кроме того, сотрудник деканата должен иметь возможность получать:

- справку о прослушанных студентом предметах с указанием часов и итоговых оценок;
- приложение к диплому выпускника также с указанием часов и итоговых оценок.

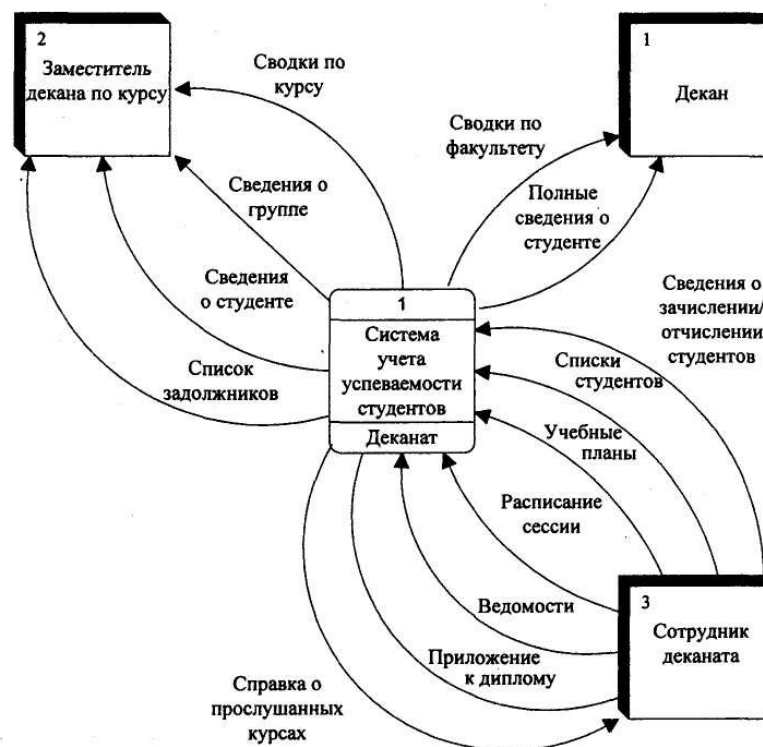


Рисунок 10 - Контекстная диаграмма системы учета успеваемости студентов (нотация Гейна-Сарсона)

Далее детализируем процессы в системе. На рисунке 11 представлена детализирующая диаграмма потоков данных, где выделены две подсистемы: *Подсистема наполнения базы данных* и *Подсистема формирования отчетов*, а также хранилище данных, которое может быть реализовано как с помощью средств СУБД, так и без них. Решение о целесообразности использования средств СУБД может быть принято позднее, после анализа структур хранимых данных.

Дальнейшую детализацию процессов можно не выполнять, так как их сущность для разработчика очевидна. Однако становится ясно, что полная спецификация данной разработки должна включать описание базы данных.



Рисунок 11 - Детализирующая диаграмма потоков данных второго уровня (нотация Гейна-Сарсона)

### Контрольные вопросы

1. Моделирование потоков данных (DFD).
2. Дайте описание компонентам DFD диаграммы: внешние сущности системы, подсистемы, процессы, накопители данных, потоки данных.
3. Опишите элементы, входящие в структуру данных (альтернатива, условное вхождение, итерация).