 Раздел 1 Аспекты разработки программного обеспечения

Тема 1.2 Анализ требований и определение спецификаций при объектном подходе.

#### Занятие 17-18 Анализ требований и определение спецификаций при объектном подходе.

Практическое занятие  - 2 часа

#### Цель: познакомить учащихся с требованиями в определении спецификаций при объектном подходе на примере использования и построения диаграмм потоков данных – DFD, систематизировать знания.

План:

#### Назначение и состав DFD.

#### [Элементы графической нотации DFD.](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#p69)

#### [Правила и рекомендации построения DFD.](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#p610)

#### [Пример построения модели DFD для системы определения допускаемых скоростей.](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#p611)

#### [Расширения DFD для систем реального времени.](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#p612)

#### [Вопросы для самопроверки.](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#question)

**1. Назначение и состав DFD**

При построении функциональной модели системы альтернативой методологии [SADT](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p62) ([IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p62)) является методология ***диаграмм потоков данных*** (Data Flow Diagrams, DFD). В отличие от [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p62), предназначенной для проектирования систем вообще, DFD предназначена для проектирования информационных систем. Ориентированность этой методологии на проектирование автоматизированных систем делает ее удобным и более выгодным инструментом при построении функциональной модели TO-BE.

Как и в [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63), основу методологии DFD составляет графический язык описания процессов. Авторами одной из первых графических нотаций DFD (1979 г.) стали Эд Йордан (Yourdon) и Том де Марко (DeMarko).

В настоящее время наиболее распространенной является нотация Гейна-Сарсона (Gane-Sarson).

Модель системы в нотации DFD представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. Модель системы содержит контекстную диаграмму и диаграммы декомпозиции.

Принципы построения функциональной модели с помощью DFD аналогичны принципам методологии [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p62). Вначале строится контекстная диаграмма, где отображаются связи системы с внешним окружением. В дальнейшем выполняется декомпозиция основных процессов и подсистем с построением иерархии диаграмм.

**2 Элементы графической нотации DFD**

Согласно DFD 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.

При построении диаграмм различают элементы графической нотации, представленные в табл. 6.1.

Таблица 6.1. Элементы графической нотации DFD

|  |  |  |
| --- | --- | --- |
| Наименование | Нотация Йордана | Нотация Гейна-Сарсона |
| Поток данных | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181892796/learning/pris/lecture/tema6/tema6_3/DFD_Iordan_Strelka.png | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181892796/learning/pris/lecture/tema6/tema6_3/DFD_Iordan_Strelka.png |
| Процесс (система, подсистема) | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181888300/learning/pris/lecture/tema6/tema6_3/DFD_Iordan_Process.png | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181876993/learning/pris/lecture/tema6/tema6_3/DFD_Gane_Process.png |
| Накопитель данных | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181884634/learning/pris/lecture/tema6/tema6_3/DFD_Iordan_Nacopitel.png | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181873607/learning/pris/lecture/tema6/tema6_3/DFD_Gane_Nacopitel.png |
| Внешняя сущность | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181880311/learning/pris/lecture/tema6/tema6_3/DFD_Iordan_Entity.png | https://sites.google.com/site/anisimovkhv/_/rsrc/1392181869624/learning/pris/lecture/tema6/tema6_3/DFD_Gane_Entity.png |

Далее в примерах будет использоваться нотация Гейна-Сарсона.

***Поток данных*** определяет информацию (материальный объект), 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.

Каждый поток данных имеет имя, отражающее его содержание. Направление стрелки показывает направление потока данных. Иногда информация может двигаться в одном направлении, обрабатываться и возвращаться назад в ее источник. Такая ситуация может моделироваться либо двумя различными потоками, либо одним – двунаправленным.

На диаграммах [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63) потоки данных соответствуют входам и выходам, но в отличие от [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63) стрелки потоков на DFD могут отображаться входящими и выходящими из любой грани внешней сущности, процесса или накопителя данных.

***Процесс*** (в [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63) – функция, работа) представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

Каждый процесс должен иметь имя в виде предложения с глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- «Ввести сведения о клиентах»;

- «Рассчитать допускаемую скорость»;

- «Сформировать ведомость допускаемых скоростей»

Номер процесса служит для его идентификации и ставится с учетом декомпозиции. В отличие от [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63) вложенность процессов обозначается через точку (например, в [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p65) – «236», в DFD – «2.3.6»).

Преобразование информации может показываться как с точки зрения процессов, так и с точки зрения ***систем*** и ***подсистем***. Если вместо имени процесса «Рассчитать допускаемую скорость» написать «Подсистема расчета допускаемых скоростей», тогда этот блок на диаграмме стоит рассматривать, как подсистему.

***Накопитель (хранилище) данных***представляет собой 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 ящика в картотеке, области в оперативной памяти, файла на магнитном носителе и т.д.

Накопителю обязательно должно даваться уникальное имя и номер в пределах всей модели (всего набора диаграмм). Имя накопителя выбирается из соображения наибольшей информативности для разработчика. Например, если в качестве накопителей выступают таблицы проектируемой базы данных, тогда в качестве имен накопителей рекомендуется использовать имена таблиц. Таким образом, накопитель данных может представлять собой всю базу данных целиком, совокупность таблиц или отдельную таблицу. Такое представление накопителей в дальнейшем облегчит построение информационной модели системы.

***Внешняя сущность*** (терминатор) представляет собой материальный объект или физическое лицо, выступающие как источник или приемник информации (например, заказчики, персонал, программа, склад, инструкция). Внешние сущности на DFD по смыслу соответствуют управлению и механизмам, отображаемым на контекстной диаграмме [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63).

Определение некоторого объекта, субъекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ проектируемой информационной системы. В связи с этим внешние сущности, как правило, отображаются только на контекстной диаграмме DFD. В процессе анализа и проектирования некоторые внешние сущности могут быть перенесены на диаграммы декомпозиции, если это необходимо, или, наоборот, часть процессов (подсистем) может быть представлена как внешняя сущность.

**3 Правила и рекомендации построения DFD**

Правила и рекомендации построения модели DFD в основном совпадают с принятыми в [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p65). Часть из них приведена в [подразд. 6.9](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2" \l "p69).

По аналогии с [IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p65) у каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.

Каждый накопитель данных также должен иметь как минимум один входящий и один выходящий поток. Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.

Наличие только выходящих потоков из накопителя также является ошибкой. Прежде чем использовать данные из накопителя, они должны там появиться в результате работы какого-либо процесса (подсистемы, внешней сущности). Исключением из правил считается случай, когда накопитель является внешней сущностью. Тогда допускается наличие либо только входящих стрелок, либо только выходящих стрелок (см. рис. [6.23](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#ris623), накопитель «БД АРМ-П или СБД-П»).

**4 Пример построения модели DFD для системы определения допускаемых скоростей**

Описание задачи приведено в [подразд. 6.6](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2" \l "p66).

Построение функциональной модели DFD начинается как и в IDEF0 с разработки контекстной диаграммы. На ней отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных. Допускается на контекстной диаграмме отображать сразу несколько основных процессов или подсистем. Пример контекстной диаграммы для рассматриваемой задачи приведен на следующем рисунке.

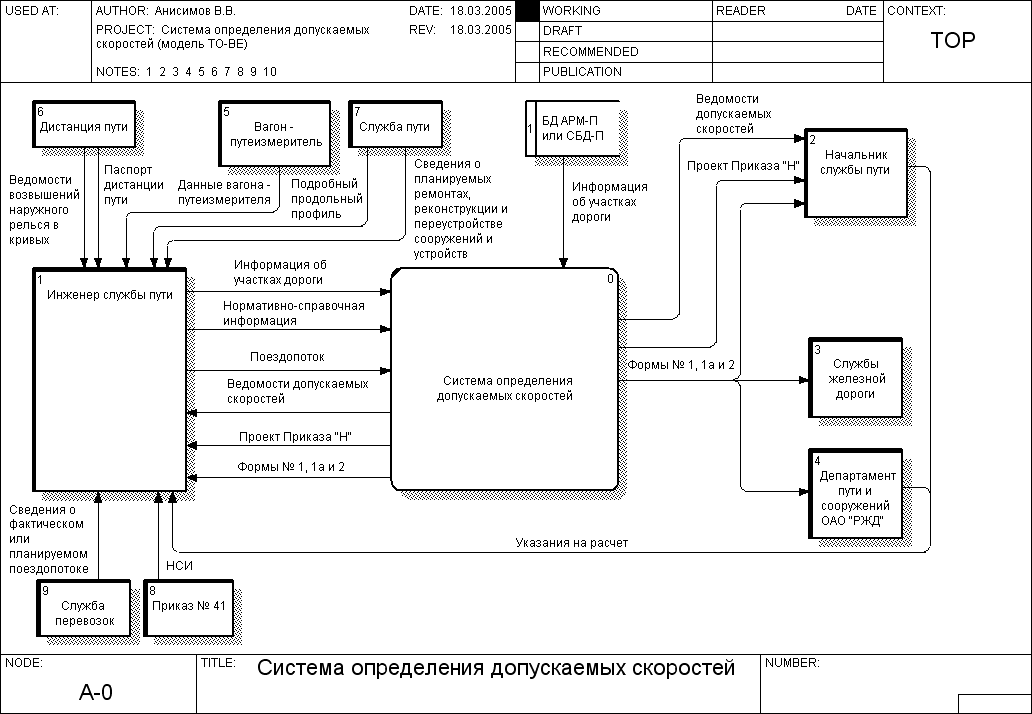


Рис. 6.23. Контекстная диаграмма системы определения допускаемых скоростей (методология DFD)

На этой диаграмме видно, что в качестве источника исходных данных для работы системы могут использоваться базы данных АРМ-П (АРМ службы пути) или СБД-П (Сводная БД – Путейский фрагмент), содержащие практически всю необходимую информацию по участкам дороги.

В то же время в системе оставлена возможность ее ручного ввода и корректировки. Несмотря на то, что БД АРМ-П или СБД-П по отношению к системе являются внешними сущностями, они, в целях лучшего восприятия, показаны в виде накопителя данных.

Дальнейший процесс проектирования состоит в построении диаграмм декомпозиции, которые строятся (показывают устройство) ***только для процессов или подсистем (систем)***.

Диаграмма декомпозиции первого уровня проектируемой системы приведена на следующем рисунке.

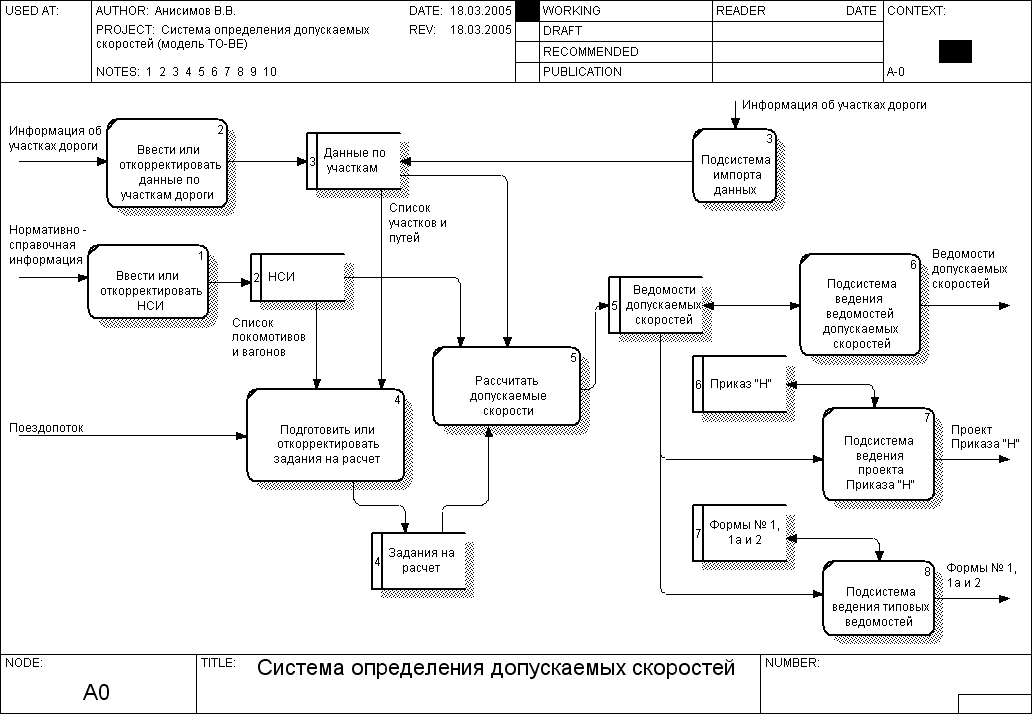


Рис. 6.24. Диаграмма декомпозиции первого уровня (методология DFD)

На этом рисунке у некоторых потоков данных, связанных с накопителями, отсутствуют имена. Это позволяет устранить дублирование надписей и, как следствие, уменьшить насыщенность диаграммы.

При построении диаграммы декомпозиции блоки системы в одних случаях показаны как процессы (имя начинается с глагола), в других – как подсистемы (имя начинается со слова «подсистема»). Это сделано в целях иллюстрации правил именования блоков. В то же время декомпозицию системы можно было бы представить, либо используя только процессы, либо только подсистемы.

Контекстная диаграмма и диаграмма декомпозиции выполнены с использованием BPwin 4.0.

Решение о завершении детализации процесса и использовании миниспецификации принимается проектировщиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2–3 потока);

- возможности описания процессов в виде простого алгоритма;

- возможности описания логики процесса при помощи миниспецификации небольшого объема (не более 20–30 строк).

Модель DFD, помимо описания функционального аспекта системы, содержит также сведения об информационном и компонентном аспектах. Совокупность накопителей данных является прообразом будущей БД, т.е. определяет состав и структуру информации. Построение диаграмм с использованием в качестве блоков подсистем показывает состав и связи компонентов будущей системы.

**5 Расширения DFD для систем реального времени**

Системы реального времени построены, как правило, на взаимодействии средств вычислительной техники и различных физических устройств съема информации (датчиков, камер, микрофонов и т. д.). Первые являются дискретными преобразователями информации, вторые в основном – аналоговыми, т.е. генерирующими информацию в виде непрерывного потока. Другой особенностью таких систем является значительный уклон в сторону управления объектами. Для моделирования особенностей поведения систем реального времени П. Вард и С. Меллор предложили использовать на DFD дополнительные элементы.

***Квазинепрерывный поток***(лат. quasi – как будто, якобы) – поток данных, непрерывный во времени. Отображается линией с двумя стрелками на конце.

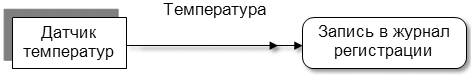


Рис. 6.25. Квазинепрерывный поток

***Управляющий процесс*** – процесс, формирующий сигналы управления на выходе.

https://sites.google.com/site/anisimovkhv/_/rsrc/1499999634873/learning/pris/lecture/tema6/tema6_3/DFD_Real_Process.png

Рис. 6.26. Управляющий процесс

***Управляющий поток*** – управляющая информация, запускающая процесс (подсистему) или изменяющая ход его выполнения.

https://sites.google.com/site/anisimovkhv/_/rsrc/1499999646896/learning/pris/lecture/tema6/tema6_3/DFD_Real_StrelkaUpr.png

Рис. 6.27. Управляющий поток

Использование управляющих потоков позволяет отделить управляющую информацию от обрабатываемой, как это делается на диаграммах IDEF0.

***Накопитель управлений*** – накопитель управляющих потоков.

https://sites.google.com/site/anisimovkhv/_/rsrc/1499999630541/learning/pris/lecture/tema6/tema6_3/DFD_Real_Nacopitel.png

Рис. 6.28. Накопитель управления

На рис. 6.29 показан пример использования новых элементов на DFD.

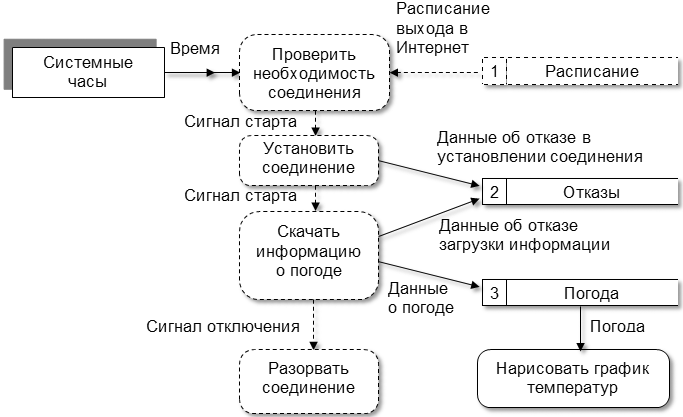


Рис. 6.29. Фрагмент DFD системы реального времени

**6 Вопросы для самопроверки**

1. [Что отображается на функциональной модели системы?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema4#KlassAspect)

2. Дайте краткую характеристику моделей [AS-IS](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_1#ASIS), [TO-BE](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_1#TOBE) и [SHOULD-BE](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_1#SHOULDBE).

3. [Перечислите виды диаграмм, используемых в IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#IDEF0Diagrams).

4. [Что показывает контекстная диаграмма?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#ContexDiagram)

5. [Перечислите элементы графической нотации IDEF0](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#Elements).

6. [Назовите назначение ICOM-кодов](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p67).

7. [Перечислите типы связей между функциями](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p64).

8. [Назовите основной принцип, определяющий объединение функций в модули](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#PrinzipRS).

9. [Назовите правила именования стрелок при их ветвлении](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#VetvlenieStrelki).

10. [Какие стрелки должны обязательно входить и выходить из блока (функции)?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#ObazStrelki)

11. [Для чего применяется механизм туннелирования стрелок?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#Tunnel)

12. [Какие связи отображаются на диаграмме дерева узлов?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#IerarxRS)

13. Перечислите отличия методологии IDEF0 от DFD.

14. [Какая модель (по отображаемому аспекту) строится с использованием диаграмм потоков данных?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#Model)

15. [Перечислите основные элементы графической нотации диаграмм потоков данных](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#ElementModel).

16. [Для каких элементов на диаграмме потоков данных можно выполнять декомпозицию?](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#Decomp)