

НОТАЦИЯ DFD

DFD — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Диаграмма потоков данных (data flow diagram, DFD) — один из основных инструментов структурного анализа и проектирования **информационных систем**, существовавших до широкого распространения UML.









Итак DFD – это нотация, предназначенная для **моделирования информационных систем** с точки зрения хранения, обработки и передачи данных, то есть используется разработчиками ИС для разработчиков ИС.

DFD должна наглядно ответить на вопросы:

- из чего состоит информационная система?
- что нужно, чтобы обработать информацию?

Зачем нужна нотация DFD?

Исторически синтаксис этой нотации применяется в двух вариантах — Йордана (Yourdon) и Гейна-Сарсона (Gane-Sarson). Различия между ними – в таблице ниже:

	Yourdon	Gane-Sarson
Внешняя сущность		
Процесс		
Хранилище данных		
Поток данных		

Непосредственно DFD нотация состоит из следующих элементов:

1. Процесс (англ. Process), т.е. функция или последовательность действий, которые нужно предпринять, чтобы данные были обработаны. Это может быть создание заказа, регистрация клиента и т.д. В названиях процессов принято использовать глаголы, т.е. «Создать клиента» (а не «создание клиента») или «обработать заказ» (а не «проведение заказа»). Здесь нет строгой системы требований, как, например, в IDEF0 или BPMN, где нотации имеют жестко определенный синтаксис, так как они могут быть исполняемыми. Но все же определенных правил стоит придерживаться, чтобы не вносить путаницу при чтении DFD другими людьми.
2. Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных. Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.

3. Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе. Поступившие данные перед обработкой и результат после обработки, а также промежуточные значения должны где-то храниться. Это и есть базы данных, таблицы или любой другой вариант организации и хранения данных. Здесь будут храниться данные о клиентах, заявки клиентов, расходные накладные и любые другие данные, которые поступили в систему или являются результатом обработки процессов.
4. Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме.

Нотация DFD может описывать любые действия, в том числе, процесс продажи или отгрузки товара, работу с заявками от клиентов или закупки материалов, с точки зрения описания системы. Но DFD не является описанием всего бизнес-процесса, она затрагивает меньше сущностей. Эта нотация помогает понять, из чего должна состоять информационная система и что нужно для автоматизации бизнес-процесса. В этой нотации описывается не столько непосредственно бизнес-процесс, сколько движение потоков данных.

Разработка модели в нотации DFD

Давайте для примера рассмотрим нотацию автоматизации продаж. Допустим, у нас есть клиент, который делает заявку через сайт или по телефону. Есть менеджер, который регистрирует эту заявку. Таким образом, в системе появляются данные – клиент и его заказ. Работник склада должен это увидеть и произвести отгрузку товара с оформлением всех необходимых документов и передать документы клиенту.

Последовательность такая:

- клиент предоставляет свои данные и заявку,
- менеджер проверяет и вносит полученные данные в систему,
- работник склада формирует документы, например, расходную накладную, и отгружает товар,
- клиент получает товар и пакет документов к нему.

Эту последовательность действий нам необходимо увидеть с точки зрения хранения данных и работы с ними в IT-системе.

С точки зрения DFD у нас имеются:

- покупатель – это внешняя сущность, которая является источником данных и получением результата,
- процесс обработки заказа (подтверждение и проводка данных в системе менеджером),
- сбор заказа на складе (после получения заявки),
- оформление отгрузки (создание необходимых документов).

Какие правила необходимо знать, чтобы создать DFD диаграмму:

Каждый процесс должен иметь хотя бы один вход и один выход. Смысл процессов здесь заключается в **обработке** именно **данных**, а потому процесс должен получить данные (входящая стрелка) и отдать куда-то после обработки (исходящая стрелка);

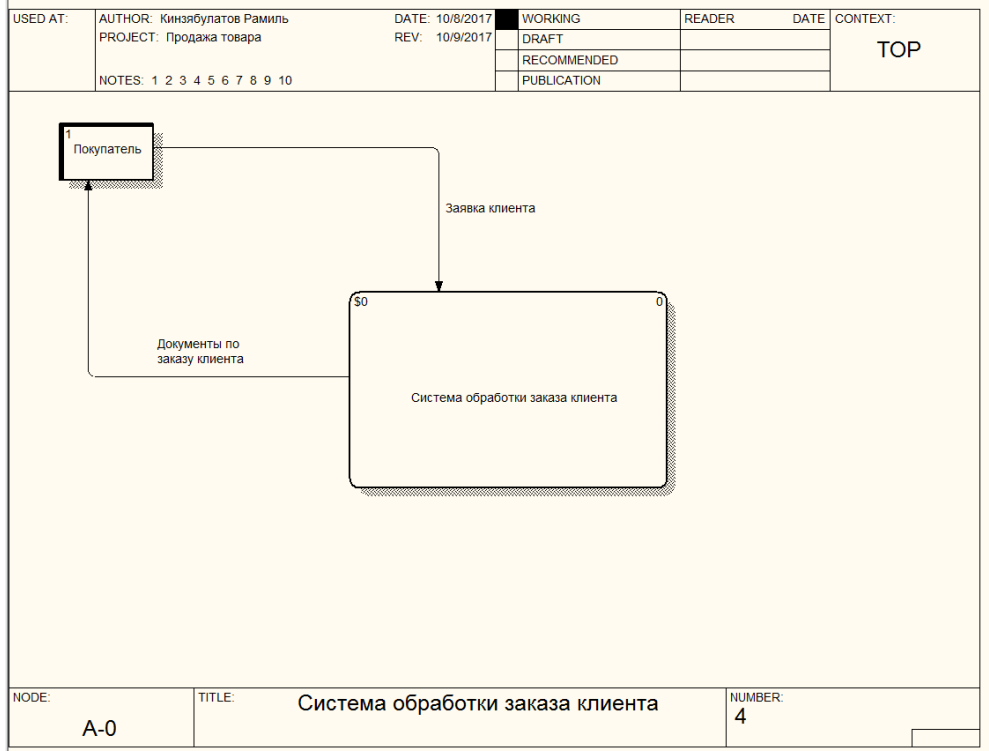
Процесс обработки данных должен иметь внешнюю входящую стрелку (данные от внешней сущности). Для того, чтобы любой подобный процесс начал работать, мало использовать данные из хранилища, должна поступить новая информация для последующей обработки;

Стрелки не могут связывать напрямую хранилища данных, все связи идут через процессы. Нет смысла просто перемещать данные из одного места в другое, а именно так читается прямая связь двух хранилищ стрелкой. Данные поступают для того, чтобы производились какие-то действия, в нашем примере – осуществлялся процесс продажи. А это возможно только посредством обработки (процесса);

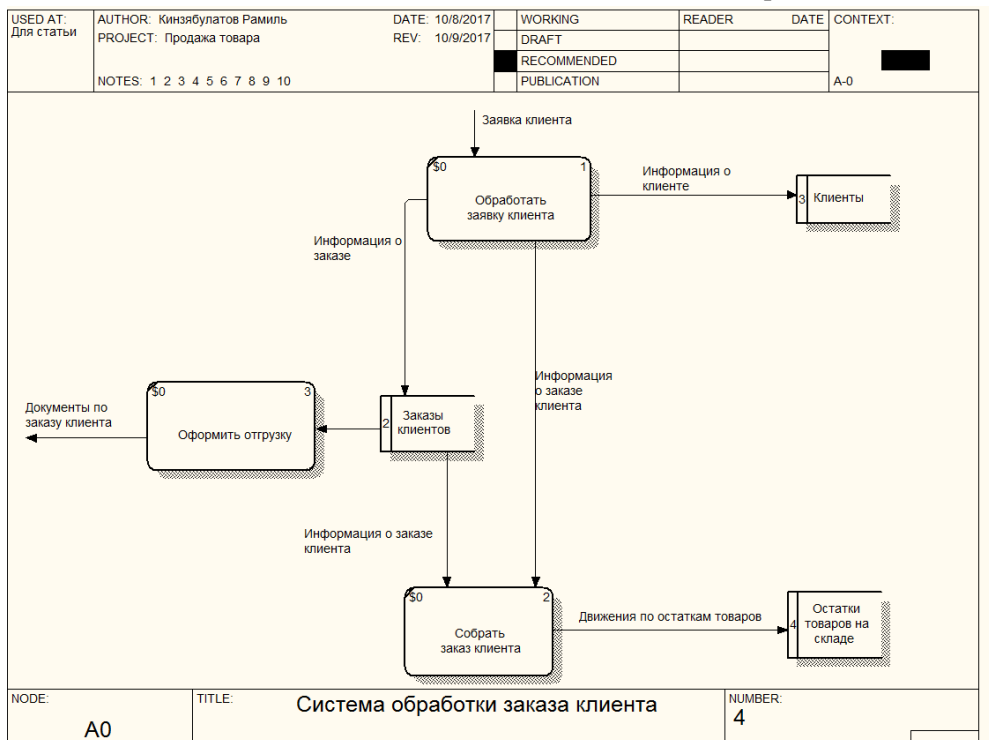
Все процессы должны быть связаны либо с другими процессами, либо с другими хранилищами данных. Процессы не существуют сами по себе, а потому результат должен куда-то передаваться;

Декомпозиция. В DFD-диаграммах предусмотрена возможность создавать крупные процессы и декомпозировать их на подпроцессы с подробным описанием действий. Например, мы можем создать процесс «создание заявки», который потом декомпозировать на последовательность действий, например, на получение заявки, отдельно – проверку и получение данных клиента, если товар в интернет-магазине продается под заказ, то также при формировании заявки потребуется получить данные от поставщика о наличии нужных наименований и т.д. И тогда на верхней диаграмме у нас будет блок «обработка заявки», а при декомпозировании мы получим диаграмму с подробной последовательностью действий на этом этапе.

Как будет выглядеть диаграмма (без декомпозиции, верхний уровень):



И декомпозиция основного элемента нашей диаграммы:



Где используются DFD нотации

DFD-диаграммы активно применяются **при разработке программного обеспечения.**

При этом:

- хранилища данных – это электронные таблицы и базы данных,
- внешние сущности – клиенты или другие базы данных, в том числе, из других программ (интеграция и обмен данными),
- процессы – это выполняемые функции и модули в системе.

Также DFD нотации удобны при анализе, когда система рассматривается с точки зрения документооборота. При этом можно наглядно увидеть, где хранятся данные, каким образом производится обмен документацией, где в этом процессе допущены ошибки организации бизнес-процессов и пр. Но здесь применение DFD диаграмм требует особой осторожности. Все же это не описание бизнес-процесса как такового, а, скорее, диаграмма перемещения данных при реализации бизнес-процессов. Но как вспомогательный вариант, в том числе, для наглядной демонстрации клиенту существующих проблем и методов оптимизации работы, этот вид нотаций вполне подойдет.

Например, для выявления проблем документооборота, дублирования документов или, наоборот, недостающей документации или электронных данных в системе, очень удобно создать отдельно – описание бизнес-процесса, а потом к нему – DFD-нотацию. Либо наоборот, предварительно для понимания основ работы бизнеса и особенностей реализации документооборота создается DFD-нотация. Она помогает выявить, например, отсутствие в системе автоматизации важных документов, которые на самом деле создаются (на бумаге), но в системе никак не отображаются. А потом уже строится оптимизированный бизнес-процесс с учетом выявленных нюансов документооборота.

В ERwin, как и в некоторых других подобных системах, существует возможность декомпозиции DFD-процессов в формате IDEF3, т.е. основная диаграмма будет в формате DFD, и на самом общем уровне вы будете видеть основные потоки данных и «узлы» их обработки. А при декомпозиции вы сможете использовать уже процессный подход, что также бывает очень удобно для разработки крупных систем или работе с разными подразделениями бизнеса.

Какое количество элементов может использоваться в DFD?

В отличие от систем с жестким синтаксисом и регламентом, в DFD нет ограничения по количеству элементов, которые могут находиться на одной диаграмме. Отсутствие ограничений дает максимум свободы и комфорта при составлении нотации, но этой свободой злоупотреблять не рекомендуется. Помните, чем больше элементов у вас на диаграмме, тем сложнее ее читать.

Имеет смысл строить DFD-диаграмму линейно, с выделением уровней функций, сущностей, хранилищ.

