

Лабораторная работа 5.

Созданной модели процессов в виде организационных диаграмм *DFD*

Цель работы: Изучить методы построения модели процессов в виде организационных диаграмм *DFD* и *Workflow (IDEF3)*.

Основные сведения



Диаграммы потоков данных (*Data Flow Diagramming*)

Диаграммы потоков данных (*Data flow diagramming, DFD*) используются для описания документооборота и обработки информации. Подобно *IDEF0*, *DFD* представляет модельную систему как сеть связанных между собой работ. Их можно использовать как дополнение к модели *IDEF0* для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации. Диаграмма потоков данных *DFD* описывает:

- 1) функции обработки информации (работы);
- 2) документы (стрелки, *arrow*), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- 3) внешние ссылки (*external references*), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- 4) таблицы для хранения документов (хранилище данных, *data store*).

В *BPwin* для построения диаграмм потоков данных используется нотация Гейна-Сарсона.

Для того чтобы дополнить модель *IDEF0* диаграммой *DFD*, нужно в процессе декомпозиции в диалоге *Activity Box Count* «кликнуть» по радиокнопке *DFD*. В палитре инструментов на новой диаграмме *DFD* появляются новые кнопки:

- 5)  - добавить в диаграмму внешнюю ссылку (*External Reference*). Внешняя ссылка является источником или приемником данных извне модели;
- 6)  - добавить в диаграмму Что описывают стрелки (*Data store*). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах.

Стрелки и объекты диаграммы *DFD*

В отличие от стрелок *IDEF0*, которые представляют собой жесткие взаимосвязи, стрелки *DFD* показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели *DFD* более похожими на физические характеристики системы – движение объектов (*data flow*), хранение объектов (*data stores*), поставка и распространение объектов (*external entities*) (рис. 5.1).

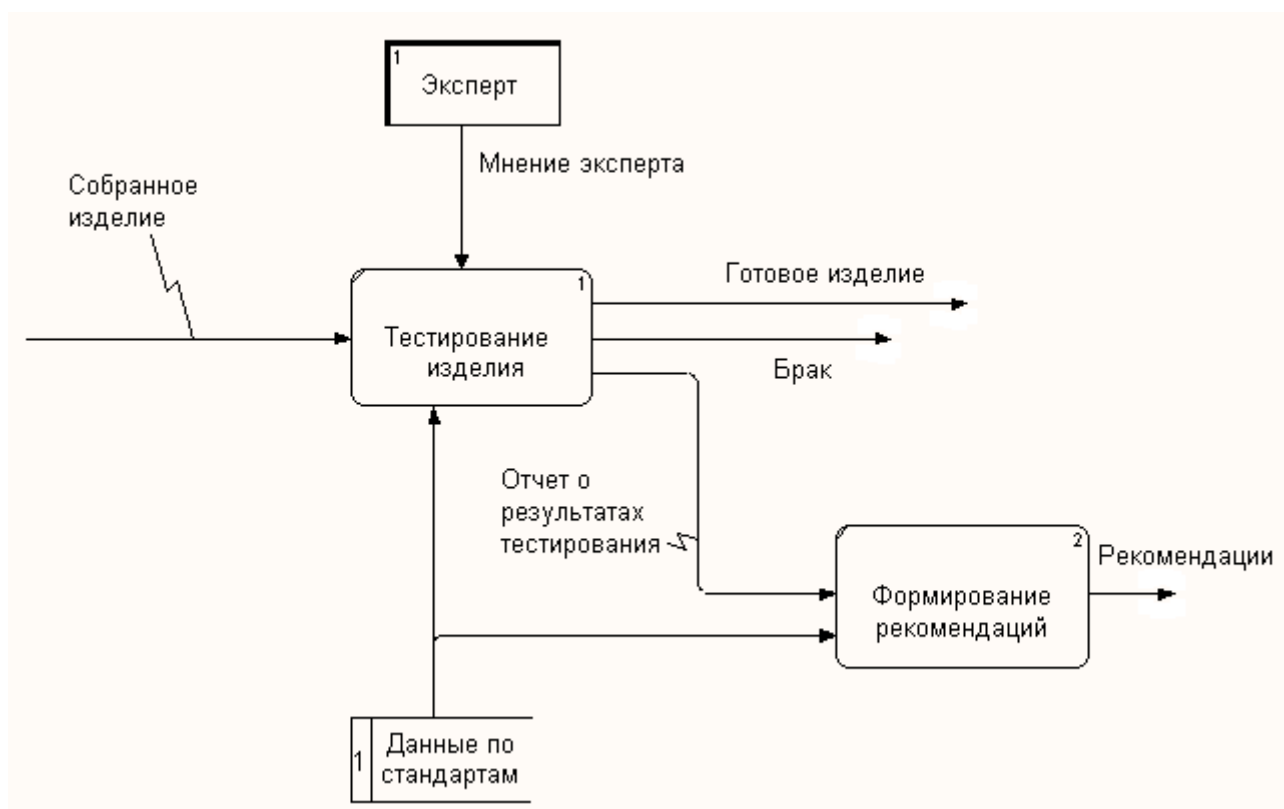


Рис. 5.1. Пример диаграммы *DFD*

В отличие от *IDEF0*, где система рассматривается как взаимосвязанные работы, *DFD* рассматривает систему как совокупность предметов. Контекстная диаграмма часто включает работы и внешние ссылки. Работы обычно именуются по названию системы, например «Система обработки информации». Включение внешних ссылок в контекстную диаграмму не отменяет требования методологии четко определить цель, область и единую точку зрения на моделируемую систему.

Работы. В *DFD* работы представляют собой функции системы, преобразующие входы в выходы. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ *IDEF0* и *IDEF3*. Так же как работы *IDEF3*, они имеют входы и выходы, но не поддерживают управления и механизмы, как *IDEF0*.

Внешние сущности. Внешние сущности изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием используют, чтобы не рисовать слишком длинных и запутанных стрелок.

Стрелки (Потоки данных). Стрелки описывают движение объектов из одной части системы в другую. Поскольку в *DFD* каждая сторона работы не имеет четкого назначения, как в *IDEF0*, стрелки могут подходить и выходить из любой грани прямоугольника работы. В *DFD* также применяются двунаправленные стрелки для описания диалогов типа «команда-ответ» между работами, между работой и внешней сущностью и между внешними сущностями (рис. 5.2).

Хранилище данных. В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое (рис. 5.3).

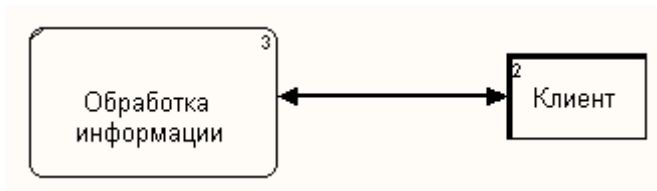


Рис. 5.2. Внешняя сущность

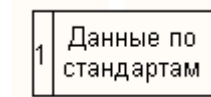


Рис. 5.3. Хранилище данных

В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов.

Слияние и разветвление стрелок. В *DFD* стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каждый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Построение диаграмм *DFD*.

Диаграммы *DFD* могут быть построены с использованием традиционного структурного анализа, подобно тому, как строятся диаграммы *IDEF0*. Сначала строится физическая модель, отображающая текущее состояние дел. Затем эта модель преобразуется в логическую модель, которая отображает требования к существующей системе. После этого строится модель, отображающая требования к будущей системе. И, наконец, строится физическая модель, на основе которой должна быть построена новая система.

Альтернативным подходом является подход, популярный при создании программного обеспечения, называемый событийным разделением (*event partitioning*), в котором различные диаграммы *DFD* выстраивают модель системы. Во-первых, логическая модель строится как совокупность работ и документирования того, что они (эти работы) должны делать.

Затем модель окружения (*environment model*) описывает систему как объект, взаимодействующий с событиями из внешних сущностей. Модель окружения обычно содержит описание цели системы, одну контекстную диаграмму и список событий. Контекстная диаграмма содержит один прямоугольник работы, изображающий систему в целом, и внешние сущности, с которыми система взаимодействует.

Наконец, модель поведения (*behavior model*) показывает, как система обрабатывает события. Эта модель состоит из одной диаграммы, в которой каждый прямоугольник изображает каждое событие из модели окружения. Хранилища могут быть добавлены для моделирования данных, которые необходимо запоминать между событиями. Потоки добавляются для связи с другими элементами, и диаграмма проверяется с точки зрения соответствия модели окружения.

Полученные диаграммы могут быть преобразованы с целью более наглядного представления системы, в частности работы на диаграммах могут быть декомпозированы.

Нумерация объектов. В *DFD* номер каждой работы может включать префикс, номер родительской работы (*A*) и номер объекта. Номер объекта – это уникальный номер работы на диаграмме. Например, работа может иметь номер *A.12.4*. Уникальный номер имеют хранилища данных и внешние сущности независимо от их расположения на диаграмме. Каждое хранилище данных имеет префикс *D* и уникальный номер, например *D5*. Каждая внешняя сущность имеет префикс *E* и уникальный номер, например *E5*.

Задание на лабораторную работу

При оформлении заказа важно проверить, существует ли такой клиент в базе данных и, если не существует, внести его в базу данных и затем оформить заказ. Оформление заказа начинается со звонка клиента. В процессе оформления заказа база данных клиентов может просматриваться и редактироваться. Заказ должен включать как информацию о клиенте, так и информацию о заказанных продуктах. Оформление заказа подразумевает чтение и запись информации о прочих заказах.

В процессе декомпозиции согласно правилам *DFD* необходимо преобразовать граничные стрелки во внутренние, начинающиеся и заканчивающиеся на внешних ссылках.

1. Декомпозируйте работу «Оформление заказов» на диаграмме A1 на две работы.
2. Для новых работ определите имена:
 - 1) «Формирование заказа»;
 - 2) «Проверка и ввод клиентов».
3. Добавьте на диаграмму следующие хранилища данных:
 - 3) «Список клиентов»;
 - 4) «Список компонентов»;
 - 5) «Список заказов».

Работа «Проверка и ввод клиентов» взаимодействует с хранилищем данных «Список клиентов» для ввода в него информации о новых клиентах и для получения информации о зарегистрированных клиентах

Работа «Формирование заказа» взаимодействует с хранилищем данных «Список клиентов» для получения информации о зарегистрированных клиентах, с хранилищем данных «Список компонентов» для получения информации об имеющихся на складе компонентах для сборки компьютеров и с хранилищем данных «Список заказов», в котором хранятся существующие заказы клиентов и запоминаются вновь сформированные заказы.

4. Добавьте на диаграмму внешнюю сущность «Звонки клиентов», которая моделирует поступающую извне информацию на вход работ «Проверка и ввод клиентов» и «Формирование заказа». Для работы «Формирование заказа» входом являются «Заявки на заказ».

5. В результате формирования *DFD* диаграммы должна получиться диаграмма, представленная на рис. 5.4.

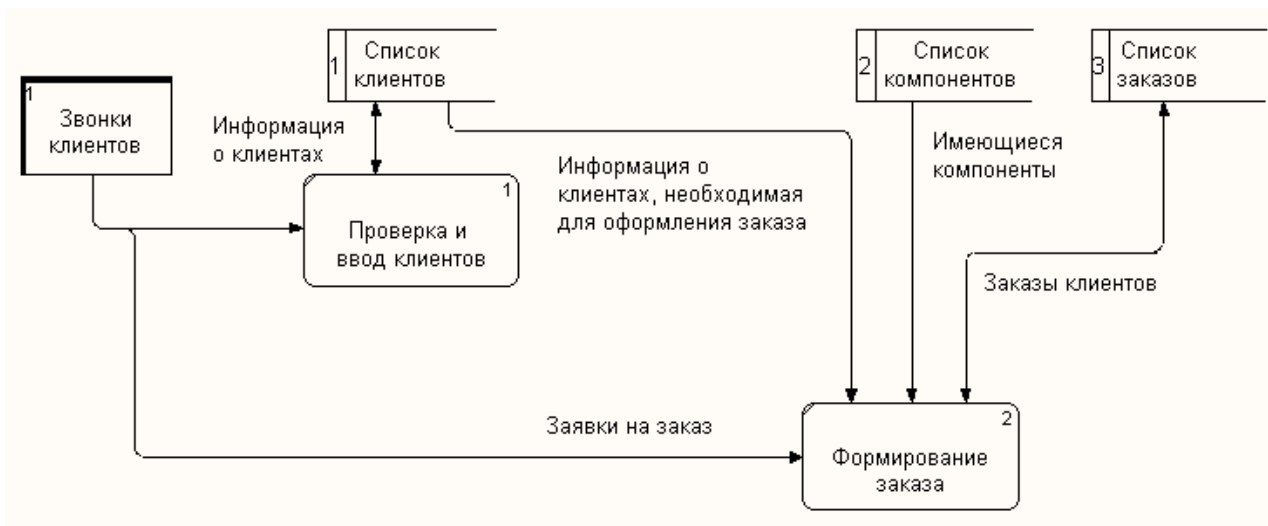


Рис. 5.4. Диаграмма декомпозиции *DFD*

Обратите внимание, что стрелки «*Информация о клиентах*» и «*Заказы клиентов*» двунаправленные. Кроме того, на родительской диаграмме *A1* стрелки, подходящие и исходящие из работы «*Оформление заказов*» станут туннелированными (рис. 5.5).

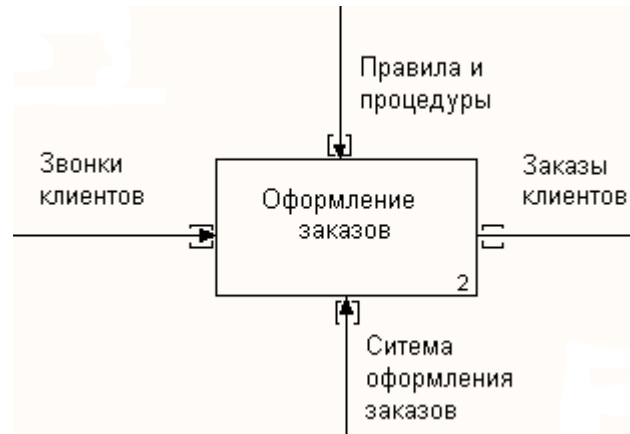


Рис. 5.5 – Работа «*Оформление заказов*» на диаграмме *A1*

Вопросы для самопроверки

1. Какое назначение имеют диаграммы *DFD*?
2. Что описывают диаграммы потоков данных *DFD*?
3. Что описывают внешние ссылки на диаграммах потоков данных *DFD*?
4. Для чего предназначены хранилища данных на диаграммах потоков данных *DFD*?
5. Что представляют работы на диаграммах потоков данных *DFD*?
6. Что описывают стрелки на диаграммах потоков данных *DFD*?
7. Для чего в диаграммах *DFD* применяются двунаправленные стрелки?