

9 ДИАГРАММЫ ПОТОКОВ ДАННЫХ

Потоки данных используются для моделирования передачи информации (или даже физических компонентов) из одной части системы в другую. Логическая диаграмма потоков данных (DFD), приведенная на рисунке 1 показывает внешние по отношению к системе источники и приемники (адресаты) данных, идентифицирует логические функции (процессы) и группы элементов данных, связывающие одну функцию с другой (потоки), а также идентифицирует хранилища (накопители) данных, к которым осуществляется доступ.



Рис. 1 Пример DFD

Структуры потоков данных и определения их компонентов хранятся и анализируются в словаре данных. Каждая логическая функция (процесс) может быть детализирована с помощью DFD нижнего уровня; когда дальнейшая детализация перестает быть полезной, переходят к выражению логики функции при помощи спецификации процесса (мини-спецификации). Содержимое каждого хранилища также сохраняют в словаре данных, модель данных хранилища раскрывается с помощью ER-диаграмм.

В частности, в DFD не показываются процессы, которые управляют собственно потоком данных и не приводятся различия между допустимыми и недопустимыми путями. DFD содержат множество полезной информации, а кроме того:

- позволяют представить систему с точки зрения данных;
- иллюстрируют внешние механизмы подачи данных, которые потребуют наличия специальных интерфейсов;
- позволяют представить как автоматизированные, так и ручные процессы системы;
- выполняют ориентированное на данные секционирование всей системы.

Потоки на диаграммах изображаются именованными стрелками, стрелки указывают направление движения информации. Иногда информация может

двигаться в одном направлении, обрабатываться и возвращаться в ее источник. Такая ситуация может моделироваться либо двумя различными потоками, либо одним двунаправленным.

Процесс преобразует входной поток данных в выходной в соответствии с действием, задаваемым именем процесса. 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.

Хранилище данных (data storage) позволяет на ряде участков определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет «срезы» потоков данных во времени. Информацию, которую оно содержит, можно использовать в любое время после ее определения, при этом данные могут выбираться в произвольном порядке. Имя хранилища должно идентифицировать его содержимое. В случае, когда поток данных входит (выходит) в (из) хранилище и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое нет необходимости отражать на диаграмме.

Внешняя сущность (терминатор) представляет сущность вне контекста системы, являющуюся источником или приемником системных данных. Ее имя должно содержать существительное, например «Клиент». Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке.

Диаграммы изменения состояний STD. Жизненный цикл сущности относится к классу STD-диаграмм ([рис. 2](#)).

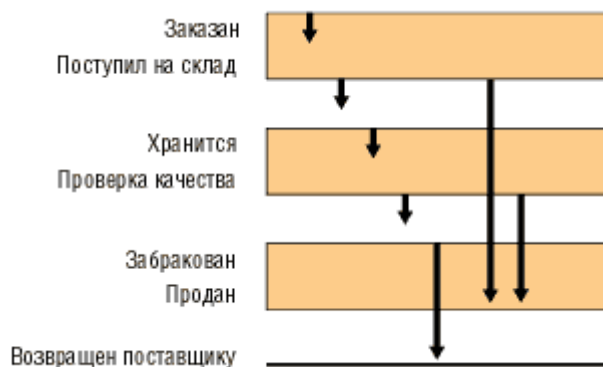


Рис. 2 Пример диаграммы жизненного цикла

Эта диаграмма отражает изменение состояния объекта с течением времени. Например, рассмотрим состояние товара на складе: товар может быть заказан у поставщика, поступить на склад, храниться на складе, проходить контроль качества, может быть продан, забракован, возвращен поставщику. Стрелки на диаграмме показывают допустимые изменения состояний.