

Глава 3. Методика IDEF0/SADT. Функциональная модель

3.1. Методология SADT/IDEF0

SADT-моделью называется описание системы с помощью SADT. В SADT-моделях используются как *естественный*, так и *графический* языки. Естественный язык служит для передачи информации о конкретной системе. При этом источником естественного языка являются люди, описывающие систему. Графический язык SADT определённым образом организует естественный язык. Источником графического языка служит сама методология SADT.

С точки зрения SADT модель может быть сосредоточена либо на *функциях* системы, либо на её *объектах*. SADT-модели, ориентированные на функции, принято называть *функциональными моделями*, а ориентированные на объекты системы — *моделями данных*.

Функциональная модель представляет с требуемой степенью детализации систему функций, которые отражают свои взаимоотношения через объекты системы. *Модели данных* представляют собой подробное описание объектов системы, связанных системными функциями. Полная методология SADT поддерживает создание множества моделей для более точного описания сложной системы.

Методология SADT/IDEF0 представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Основные элементы методологии SADT/IDEF0 основываются на следующих концепциях:

- графическое представление блочного моделирования — графика блоков и дуг SADT-диаграммы — отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него;
- строгость и точность — выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика;
- связность диаграмм — блоки нумеруются специальным образом;
- уникальность меток и наименований — отсутствие повторяющихся имён;
- синтаксические правила для графических символов (блоков и стрелок);
- разделение входов и управлений — правило определения роли данных;
- отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель.

Методология SADT/IDEF0 может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем SADT/IDEF0 может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

3.1.1. Методологические понятия

Одним из основных понятий стандарта IDEF0 является *декомпозиция (Decomposition)*. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели. Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм.

Модель IDEF0 всегда начинается с представления системы как единого целого — одного функционального блока с интерфейсными дугами, уходящими

ми за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется *контекстной диаграммой* и обозначается идентификатором «A-0».

В пояснительном тексте к контекстной диаграмме должна быть указана *цель* (*Purpose*) построения диаграммы в виде краткого описания и зафиксирована *точка зрения* (*Viewpoint*). Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь. Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Чёткое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется *дочерней* (*Child Diagram*) по отношению к нему, а каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется *дочерним блоком* (*Child Box*). В свою очередь, функциональный блок-предок называется *родительским блоком* по отношению к дочерней диаграмме (*Parent Box*), а диаграмма, к которой он принадлежит, — *родительской диаграммой* (*Parent Diagram*). Каждая из подфункций дочерней диаграммы может быть далее детализирована путём аналогичной декомпозиции соответствующего ей функционального блока. В случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме.

Каждый блок имеет свой уникальный порядковый номер на диаграмме (цифра в правом нижнем углу прямоугольника), а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы. Отсутствие

этого обозначения говорит о том, что декомпозиции для данного блока не существует.

Для каждого из элементов IDEF0 (диаграмм, функциональных блоков, интерфейсных дуг) существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображённый данным элементом. Этот набор называется *глосарием (Glossary)* и является описанием сущности данного элемента.

3.1.2. Точка зрения

С методической точки зрения при моделировании полезно использовать мнения экспертов, имеющих разные взгляды на предметную область. Однако при разработке каждой модели должна существовать единственная, заранее определённая точка зрения.

Обычно наименованием точки зрения является название должности или подразделения.

Может оказаться необходимым построение моделей с разных точек зрения для детального отражения всех особенностей моделируемой системы. Тогда в прикреплённых диаграммах кратко документируют и другие точки зрения.

3.1.3. Иерархия диаграмм

Построение IDEF0-модели начинается с представления всей системы в виде простейшей компоненты — одного блока и стрелок, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок представляет всю систему как единое целое, то имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг — они также представляют полный набор внешних интерфейсов системы в целом.

Далее блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединён-

ных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть разбита подобным образом для более детального представления.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы, так как родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Модель IDEF0 представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые представлены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции более общая диаграмма называется родительской для более детальной диаграммы.

Стрелки, входящие в блок и выходящие из него, на диаграмме верхнего уровня являются теми же, что и стрелки, входящие в диаграмму нижнего уровня и выходящие из неё, потому что блок и диаграмма представляют одну и ту же часть системы. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.

3.1.4. Принципы ограничения сложности IDEF0-диаграмм

Обычно IDEF0-модели несут в себе сложную и концентрированную информацию. Поэтому, чтобы ограничить их перегруженность и сделать удобочитаемыми, в стандарте приняты соответствующие ограничения сложности:

- ограничение количества функциональных блоков на каждом уровне декомпозиции тремя–шестью (верхний предел (шесть) заставляет разра-

ботчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать её создание);

- ограничение количества подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг четырьмя.

3.1.5. Дисциплина групповой работы над разработкой IDEF0-модели

Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большой группой людей, принадлежащих к разным областям деятельности моделируемой системы. Обычно процесс разработки является итеративным и состоит из следующих условных этапов.

- *Создание модели группой специалистов, относящихся к различным сферам деятельности предприятия, называемых авторами (Authors).* Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов. На основе имеющихся положений, документов и результатов опросов создаётся черновик (Model Draft) модели.
- *Распространение черновика для рассмотрения, согласований и комментариев.* На этой стадии происходит обсуждение черновика модели с широким спектром компетентных лиц (читателями). При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передаётся автору. Автор в свою очередь также письменно соглашается с критикой или отвергает её с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.
- *Официальное утверждение модели.* Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у

авторов модели и читателей отсутствуют разногласия по поводу её адекватности. Окончательная модель является согласованным представлением о системе с заданной точки зрения и для заданной цели.

3.2. Синтаксис и семантика моделей SADT/IDEF0

Нотация IDEF0 крайне проста. Она содержит только две сущности — блоки и стрелки.

Функциональные блоки (Activity Box) задают действия. Функциональный блок графически изображается в виде прямоугольника. Он задаёт некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, «проверить документацию», а не «проверка документации»).

Для выполнения действия могут потребоваться входные данные (сырьё, информация и т.д.). В результате мы получаем что-либо на выходе.

3.2.1. Интерфейсные дуги

Потоки информации обозначаются *интерфейсными дугами (называемые также потоками или стрелками) (Arrow)*. Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображённую данным функциональным блоком (см. рис. 3.1).

Графическим отображением интерфейсной дуги является односторонняя стрелка. Каждая интерфейсная дуга должна иметь своё *的独特ое наименование (Arrow Label)*. По требованию стандарта, наименование должно быть обратом существительного. Началом и концом каждой функциональной дуги могут быть только функциональные блоки, при этом источником может быть только выходная сторона блока, а приёмником — любая из трёх остав-

шихся. Каждый функциональный блок должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую.

Типизацию категорий информации можно описать аббревиатурой ICOM:

- I (Input), вход** — то, что потребляется в ходе выполнения процесса;
- C (Control), управление** — ограничения и инструкции, влияющие на выполнение процесса;
- O (Output), выход** — то, что является результатом выполнения процесса;
- M (Mechanism), исполняющий механизм** — то, что используется для выполнения процесса, но остаётся неизменным.

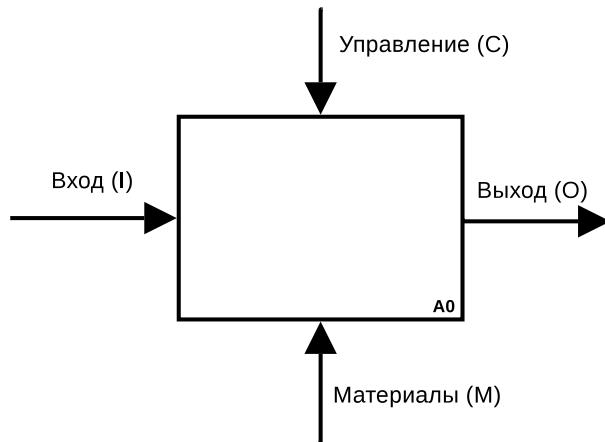


Рис. 3.1. Функциональный блок и интерфейсные дуги

Стрелки входа указывают на сырьё или информацию, потребляемые или преобразуемые функциональным блоком для производства чего-либо на выходе. Поскольку возможно существование блока, ничего не преобразующего и не изменяющего, то наличие входных стрелок не является обязательным. Стрелки входа направлены в левую сторону прямоугольника.

Стрелки управления отвечают за управлением тем, когда и как выполняется функциональный блок. Поскольку управление контролирует поведение функционального блока при создании чего-либо на выходе, то как минимум одна стрелка управления должна присутствовать у каждого блока. Стрелка управления направлена в верхнюю сторону прямоугольника.

Управление остаётся неизменным при работе блока. Если же некая инструкция или правило должно быть изменено блоком, то соответствующую информацию следует рассматривать не как управление, а как входные данные. В случае, когда неясно, относить стрелку к входу или к управлению, следует отнести её к управлению, вплоть до разрешения неясности.

Стрелки выхода указывают на наличие продукции или информации, получаемых в результате работы функционального блока. У каждого блока должен быть хотя бы один выход. Стрелки выхода направлены из правой стороны прямоугольника.

Стрелки механизма исполнения указывают на ресурсы, которые непосредственно выполняют моделируемое действие (например, персонал, техника, оборудование). Они могут отсутствовать, если не являются необходимыми для достижения поставленной цели моделирования. Стрелки механизма исполнения направлены в нижнюю сторону прямоугольника.

3.2.2. Комбинированные стрелки

Выделяют пять основных видов комбинированных стрелок: выход–вход, выход–управление, выход–механизм исполнения, выход–обратная связь на управление, выход–обратная связь на вход.

Стрелка *выход–вход* применяется, когда один из блоков должен полностью завершить работу перед началом работы другого блока (рис. 3.2).



Рис. 3.2. Комбинированная стрелка выход–вход

Стрелка *выход–управление* показывает, что один блок управляет работой другого (рис. 3.3).



Рис. 3.3. Комбинированная стрелка выход–управление

Стрелки *выход–механизм исполнения* показывают, что выход одного функционального блока применяется в качестве инструментария для работы второго (рис. 3.4).



Рис. 3.4. Комбинированная стрелка выход–механизм исполнения

Стрелка *выход–обратная связь на управление* применяется в случае, когда зависимый блок корректирует выполнение управляющего блока (рис. 3.5).

Стрелка *выход–обратная связь на вход* обычно применяется для описания циклов повторной обработки чего-либо (рис. 3.6). Стрелка изображается под блоком. Кроме того, возможно применение данной связи при повторном использовании бракованной продукции.

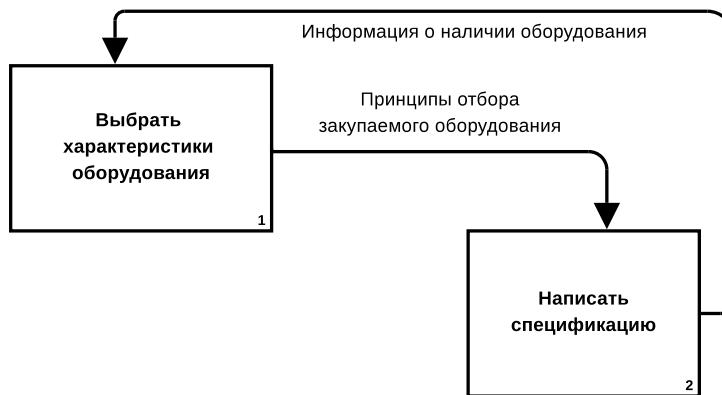


Рис. 3.5. Комбинированная стрелка выход–обратная связь на управление

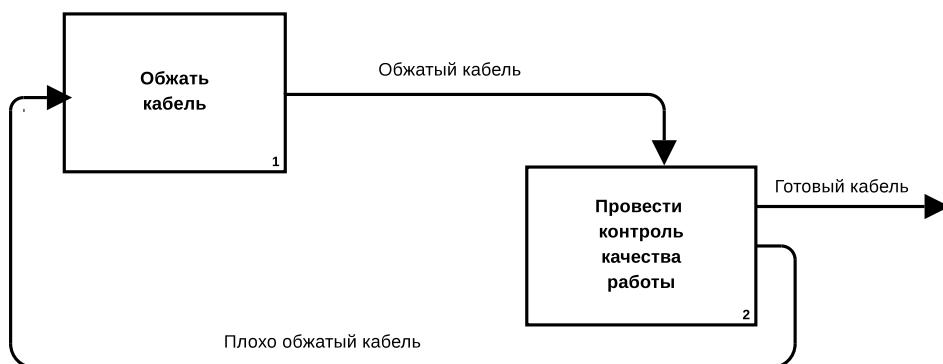


Рис. 3.6. Комбинированная стрелка выход–обратная связь на вход

3.2.3. Разъединение и соединение стрелок

Выход функционального блока может быть использован в нескольких блоках. В IDEF0 предусматривается соединение и разъединение стрелок. Разъединённые или объединённые стрелки могут иметь наименования, отличающиеся от наименования исходной стрелки. Совокупность исходной и разъединённых или объединённых стрелок называется *связанными стрелками*. Эта техника применяется для того, чтобы отразить использование только части сырья или информации, обозначаемых исходной стрелкой (рис. 3.7).

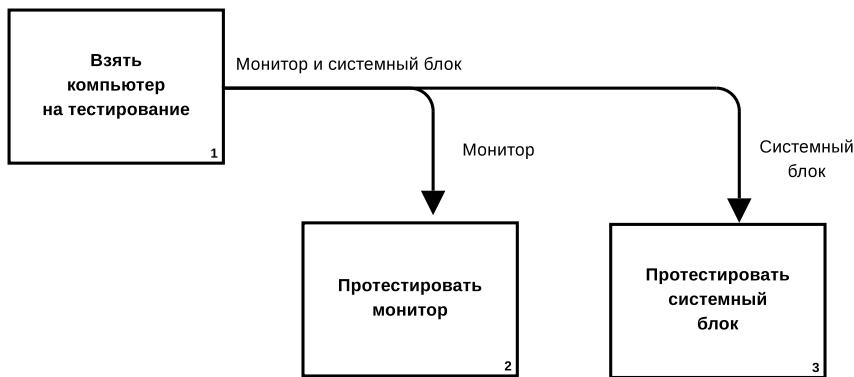


Рис. 3.7. Разъединение и переименование стрелок

3.2.4. Туннели

По методике все элементы, присутствующие на вышележащих диаграммах, должны присутствовать и на нижележащих. Но это может загромождать разработку излишними подробностями. Для управления уровнем детализации используются *туннели*. Если одна из стрелок отсутствует на родительской диаграмме (обычно в связи с несущественностью на определённом уровне абстракции) и не связана с другими стрелками родительской диаграммы, то точка входа или выхода этой стрелки обозначается туннелем. Графически это обозначается обрамлением соответствующего конца стрелки круглыми скобками. Стрелка, выходящая из туннеля, называется *стрелкой импорта ресурсов* (рис. 3.8).

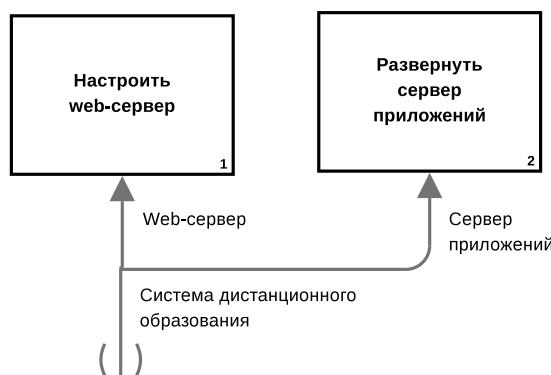


Рис. 3.8. Стрелка, выходящая из туннеля

Стрелка, входящая в туннель, называется *стрелкой подразумевания ресурса* (рис. 3.9).

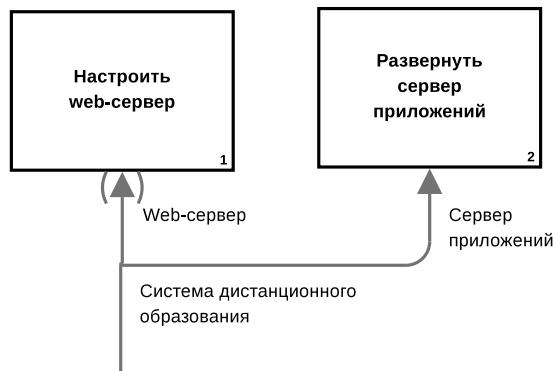


Рис. 3.9. Стрелка, входящая в туннель

3.3. Пример. Метамодель IDEF0

В качестве примера опишем метамодель IDEF0, то есть обозначим элементы диаграммы наиболее общими понятиями и проведём декомпозицию [11]. Для уменьшения объёма составим только контекстную диаграмму и её декомпозицию.

Вначале рисуется контекстная диаграмма, состоящая, как правило, из одного блока (рис. 3.10). Здесь изображены все элементы в общем виде.

Входной поток разбит на группы:

- материальные потоки;
- информационные потоки;
- энергетический поток;
- финансовый поток.

Аналогично разбит и выходной поток:

- материальные потоки;
- информационные потоки;
- финансовый поток.

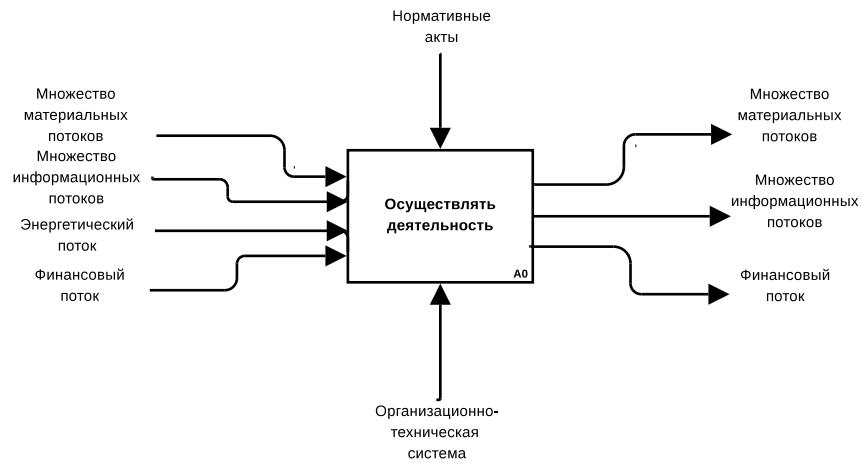


Рис. 3.10. Контекстная диаграмма

Далее происходит первая декомпозиция — декомпозиция контекстной диаграммы (рис. 3.11). На ней представлены основные действия, которые должны быть выполнены для реализации модели.

Далее производится декомпозиция каждой диаграммы в отдельности. Представляем читателю возможность это сделать самостоятельно.

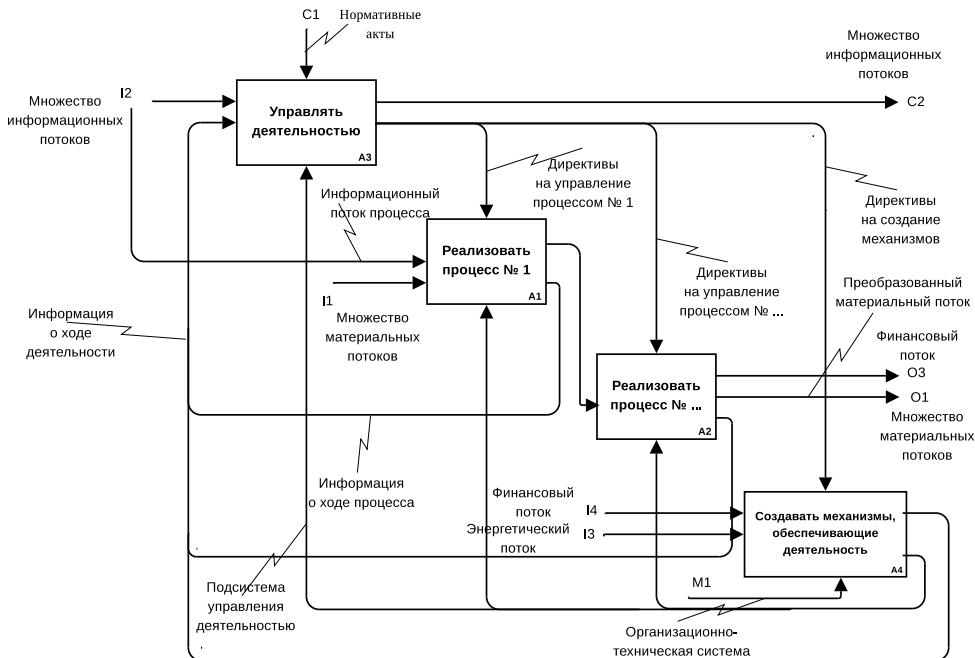


Рис. 3.11. Декомпозиция контекстной диаграммы