***МДК 03.01 Проектирование и дизайн информационных систем***

**Лекция «ПРИНЦИПЫ ПОСТРОЕНИЯ МОДЕЛИ IDEF0»**

***Методология SADT*** (Structured Analysis and Design Technique – **методология структурного анализа и проектирования) представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы.**

Начало разработки данной методологии было положено Дугласом Россом (США) в середине 60-х гг. ХХ в. С тех пор системные аналитики компании SofTech, Inc. улучшили SADT и использовали ее в решении широкого круга проблем. Программное обеспечение телефонных сетей, диагностика, долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, конфигурация компьютерных систем, обучение персонала, управление финансами и материально-техническим снабжением – вот некоторые из областей эффективного применения SADT. Широкий спектр областей указывает на универсальность и мощь методологии SADT. В программе «Интеграции компьютерных и промышленных технологий» (Integrated Computer Aided Manufacturing, ICAM) Министерства обороны США была признана полезность SADT. Это привело к публикации ее части в 1981 г., называемой ***IDEF0*** (Icam DEFinition), в качестве федерального стандарта на разработку программного обеспечения. Под этим названием SADT стала применяться тысячами специалистов в военных и промышленных организациях [[15](https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/literatura#lit15)]. **Последняя редакция стандарта IDEF0 была выпущена в декабре 1993г.** Национальным институтом по стандартам и технологиям США (National Institute Standards and Technology, NIST).

Стоит отметить, что IDEF0 рекомендована для использования Госстандартом РФ и активно применяется в отечественных госструктурах (например, в Государственной налоговой инспекции РФ).

Данная методология при описании функционального аспекта информационной системы конкурирует с методами, ориентированными на потоки данных (DFD). В отличие от них **IDEF0 позволяет:**

- **описывать любые системы, а не только информационные** (DFD предназначена для описания программного обеспечения);

**- создать описание системы и ее внешнего окружения до определения окончательных требований к ней.** Иными словами, с помощью данной методологии можно постепенно выстраивать и анализировать систему даже тогда, когда трудно еще представить ее воплощение.

Таким образом, IDEF0 может применяться на ранних этапах создания широкого круга систем. В то же время она может быть использована для анализа функций существующих систем и выработки решений по их улучшению.

**Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм.** **Каждая диаграмма является единицей описания системы и располагается на отдельном листе.**

Модель (AS-IS, TO-BE или SHOULD-BE) может содержать ***4 типа диаграмм*** :

**- контекстную диаграмму;**

**- диаграммы декомпозиции;**

**- диаграммы дерева узлов;**

**- диаграммы только для экспозиции (for exposition only, FEO).**

***Контекстная диаграмма*(диаграмма верхнего уровня), являясь вершиной древовидной структуры диаграмм, показывает назначение системы (основную функцию) и ее взаимодействие с внешней средой.** В каждой модели может быть только одна контекстная диаграмма. После описания основной функции выполняется функциональная декомпозиция, т. е. определяются функции, из которых состоит основная.

**Далее функции делятся на подфункции и так до достижения требуемого уровня детализации исследуемой системы**. Диаграммы, которые описывают каждый такой фрагмент системы, называются ***диаграммами декомпозиции***. **После каждого сеанса декомпозиции проводятся сеансы экспертизы – эксперты предметной области указывают на соответствие реальных процессов созданным диаграммам.** Найденные несоответствия устраняются, после чего приступают к дальнейшей детализации процессов.

***Диаграмма дерева узлов*** показывает **иерархическую зависимость функций (работ), но не связи между ними.** Их может быть несколько, поскольку дерево можно построить на произвольную глубину и с произвольного узла.

***Диаграммы для экспозиции*строятся для иллюстрации отдельных фрагментов модели с целью отображения альтернативной точки зрения на происходящие в системе процессы (например, с точки зрения руководства организации).**

**Элементы графической нотации IDEF0**

Методология IDEF0 нашла широкое признание и применение, в первую очередь, благодаря простой графической нотации, используемой для построения модели. Главными компонентами модели являются диаграммы. На них отображаются функции системы в виде прямоугольников, а также связи между ними и внешней средой посредством стрелок. Использование всего лишь двух графических примитивов (прямоугольник и стрелка) позволяют быстро объяснить правила и принципы построения диаграмм IDEF0 людям, незнакомым с данной методологией. Это достоинство позволяет подключить и активизировать деятельность заказчика по описанию бизнес-процессов с использованием формального и наглядного графического языка.

На следующем рисунке показаны основные элементы графической нотации IDEF0.

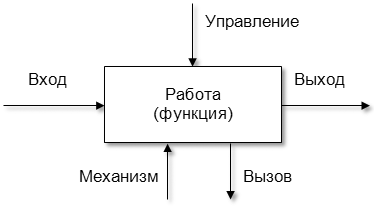


Рис. 6.1. Элементы графической нотации IDEF0

Прямоугольник представляет собой ***работу (процесс, деятельность, функцию или задачу)***, которая имеет фиксированную цель и приводит к некоторому конечному результату. Имя работы должно выражать действие (например, «Изготовление детали», «Расчет допускаемых скоростей», «Формирование ведомости ЦДЛ № 3»).

Взаимодействие работ между собой и внешним миром описывается в виде стрелок. В IDEF0 различают ***5 видов стрелок***:

- ***вход*** (англ. input) – **материал или информация, которые используются и преобразуются работой для получения результата (выхода).** Вход отвечает на вопрос «Что подлежит обработке?». В качестве входа может быть как материальный объект (сырье, деталь, экзаменационный билет), так и не имеющий четких физических контуров (запрос к БД, вопрос преподавателя). Допускается, что работа может не иметь ни одной стрелки входа**. Стрелки входа всегда рисуются входящими в левую грань работы;**

- ***управление*** (англ. control) – **управляющие, регламентирующие и нормативные данные, которыми руководствуется работа. Управление отвечает на вопрос «В соответствии с чем выполняется работа?».** Управление влияет на работу, но не преобразуется ей, т.е. выступает в качестве ограничения. **В качестве управления могут быть правила, стандарты, нормативы, расценки, устные указания. Стрелки управления рисуются входящими в верхнюю грань работы. Если при построении диаграммы возникает вопрос, как правильно нарисовать стрелку сверху или слева, то рекомендуется ее рисовать как вход (стрелка слева);**

**- *выход* (англ. output) – материал или информация, которые представляют результат выполнения работы. Выход отвечает на вопрос «Что является результатом работы?». В качестве выхода может быть как материальный объект (деталь, автомобиль, платежные документы, ведомость), так и нематериальный (выборка данных из БД, ответ на вопрос, устное указание). Стрелки выхода рисуются исходящими из правой грани работы;**

**- *механизм* (англ. mechanism) – ресурсы, которые выполняют работу. Механизм отвечает на вопрос «Кто выполняет работу или посредством чего?». В качестве механизма могут быть персонал предприятия, студент, станок, оборудование, программа. Стрелки механизма рисуются входящими в нижнюю грань работы;**

**- *вызов* (англ. call) – стрелка указывает, что некоторая часть работы выполняется за пределами рассматриваемого блока. Стрелки выхода рисуются исходящими из нижней грани работы.**

**Типы связей между работами**

 После определения состава функций и взаимосвязей между ними, возникает вопрос о правильной их композиции (объединении) в модули (подсистемы). При этом подразумевается, что каждая отдельная функция должна решать одну, строго определенную задачу. В противном случае необходима дальнейшая декомпозиция или разделение функций.

При объединении функций в подсистемы необходимо стремиться, чтобы внутренняя связность (между функциями внутри модуля) была как можно сильнее, а внешняя (между функциями, входящими в разные модули), как можно слабее. Опираясь на семантику связей методологии [IDEF0](https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2#p63), введем классификацию связей между функциями (работами). Данная классификация является расширением [[14](https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/literatura#lit14)]. Типы связей приводятся в порядке уменьшения их значимости (силы связывания). В приводимых примерах утолщенными линиями выделяются функции, между которыми имеется рассматриваемый тип связи.

1. ***Иерархическая связь (связь «часть» – «целое»)*** **имеет место между функцией и подфункциями, из которых она состоит.**

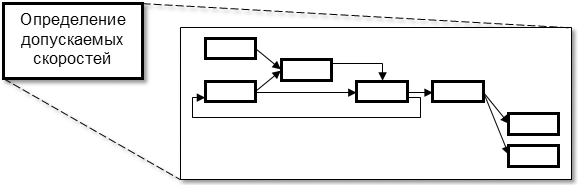


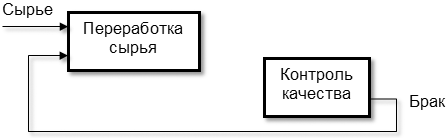
Рис. 6.2. Иерархическая связь

2. ***Регламентирующая (управляющая, подчиненная) связь*** **отражает зависимость одной функции от другой, когда выход одной работы направляется на управление другой. Функцию, из которой выходит управление, следует считать регламентирующей или управляющей, а в которую входит – подчиненной. Различают *прямую связь по управлению*, когда управление передается с вышестоящей работы на нижестоящую (рис. 6.3), и *обратную связь по управлению*, когда управление передается от нижестоящей к вышестоящей (рис. 6.4).**

|  |  |
| --- | --- |
| https://www.sites.google.com/site/anisimovkhv/_/rsrc/1499998302299/learning/pris/lecture/tema6/tema6_2/IDEF0_Relationship_UprPram.png | https://www.sites.google.com/site/anisimovkhv/_/rsrc/1499998298301/learning/pris/lecture/tema6/tema6_2/IDEF0_Relationship_UprObr.png |
| Рис. 6.3. Прямая связь по управлению | Рис. 6.4. Обратная связь по управлению |

3. ***Функциональная (технологическая) связь*** **имеет место, когда выход одной функции служит входными данными для следующей функции.** С точки зрения потока материальных объектов данная связь показывает технологию (последовательность работ) обработки этих объектов. **Различают *прямую связь по входу*, когда выход передается с вышестоящей работы на нижестоящую (рис. 6.5), и *обратную связь по входу*, когда выход передается с нижестоящей к вышестоящей (рис.6.6).**

|  |  |
| --- | --- |
| https://www.sites.google.com/site/anisimovkhv/_/rsrc/1499998262293/learning/pris/lecture/tema6/tema6_2/IDEF0_Relationship_FunPram.png |  |
| Рис. 6.5. Прямая связь по входу |  |

 Рис. 6.6. Обратная связь по входу

4. ***Потребительская связь*** **имеет место, когда выход одной функции служит механизмом для следующей функции. Таким образом, одна функция потребляет ресурсы, вырабатываемые другой.**

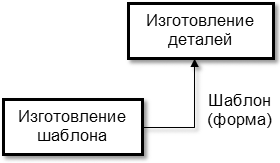


Рис. 6.7. Потребительская связь

5. ***Логическая связь*** **наблюдается между логически однородными функциями. Такие функции, как правило, выполняют одну и ту же работу, но разными (альтернативными) способами или, используя разные исходные данные (материалы).**

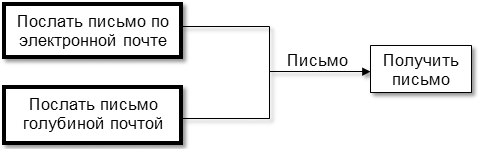


Рис. 6.8. Логическая связь

6. ***Коллегиальная (методическая) связь*** **имеет место между функциями, алгоритм работы которых определяется одним и тем же управлением.** Аналогом такой связи является совместная работа сотрудников одного отдела (коллег), подчиняющихся начальнику, который отдает указания и приказы (управляющие сигналы). Такая связь также возникает, когда алгоритмы работы этих функций определяются одним и тем же методическим обеспечением (СНИП, ГОСТ, официальными нормативными материалами и т. д.), служащим в качестве управления.

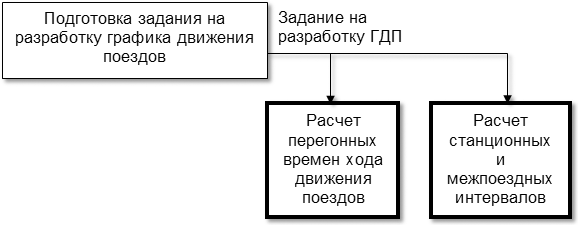


Рис. 6.9. Методическая связь

7. ***Ресурсная связь*** **возникает между функциями, использующими для своей работы одни и те же ресурсы. Ресурсно-зависимые функции, как правило, не могут выполняться одновременно.**

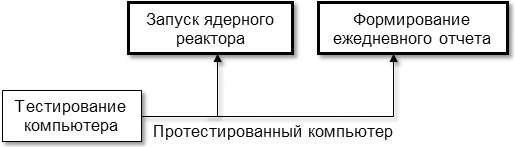


Рис. 6.10. Ресурсная связь

8. ***Информационная связь*** **имеет место между функциями, использующими в качестве входных данных одну и ту же информацию.**

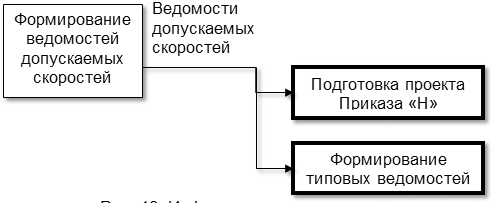


Рис. 6.11. Информационная связь

9. ***Временная связь*** **возникает между функциями, которые должны выполняться одновременно до или одновременно после другой функции.**

Кроме указанных на рисунке случаев, эта связь имеет место также между другими сочетаниями управления, входа и механизма, поступающими в одну функцию.

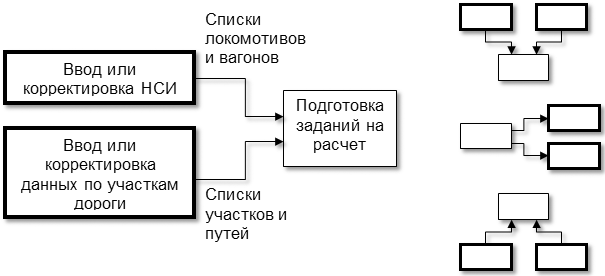


Рис. 6.12. Временная связь

10. ***Случайная связь*** **возникает, когда конкретная связь между функциями мала или полностью отсутствует.**



Рис. 6.13. Случайная связь

Из приведенных выше типов связей наиболее сильной является иерархическая связь, которая, по сути, и определяет объединение функций в модули (подсистемы). Несколько слабее являются регламентирующие, функциональные и потребительские связи. Функции с этими связями обычно реализуются в одной подсистеме. Логические, коллегиальные, ресурсные и информационные связи одни из самых слабых. Функции, обладающие ими, как правило, реализуют в разных подсистемах, за исключением логически однородных функций (функций, связанных логической связью). Временная связь свидетельствует о слабой зависимости функций друг от друга и требует их реализации в отдельных модулях.

Таким образом, при объединении функций в модули наиболее желательными являются первые пять видов связей. Функции, связанные последними пятью связями, лучше реализовывать в отдельных модулях.

**Правила и рекомендации построения диаграмм IDEF0**

В IDEF0 существуют соглашения (правила и рекомендации) по созданию диаграмм, которые призваны облегчить чтение и экспертизу модели. Некоторые из этих правил CASE-средства поддерживают автоматически, выполнение других следует обеспечить вручную.

1. Перед построением модели необходимо определиться, какая модель (модели) системы будет построена. Это подразумевает определение ее типа AS-IS, TO-BE или SHOULD-BE, а также определения позиции, с точки зрения которой строится модель. «Точку зрения» лучше всего представлять себе как место (позицию) человека или объекта, в которое надо встать, чтобы увидеть систему в действии. Например, при построении модели работы продуктового магазина можно среди возможных претендентов, с точки зрения которых рассматривается система, выбрать продавца, кассира, бухгалтера или директора. Обычно выбирается одна точка зрения, наиболее полно охватывающая все нюансы работы системы, и при необходимости для некоторых диаграмм декомпозиции строятся диаграммы FEO, отображающие альтернативную точку зрения.

2. На контекстной диаграмме отображается один блок, показывающий назначение системы. Для него рекомендуется отображать по 2–4 стрелки, входящие и выходящие с каждой стороны.

3. Количество блоков на диаграммах декомпозиции рекомендуется в пределах 3–6. Если на диаграмме декомпозиции два блока, то она, как правило, не имеет смысла. При наличии большого количества блоков диаграмма становится перенасыщенной и трудно читаемой.

4. Блоки на диаграмме декомпозиции следует располагать слева направо и сверху вниз. Такое расположение позволяет более четко отразить логику и последовательность выполнения работ. Кроме этого маршруты стрелок будут менее запутанными и иметь минимальное количество пересечений.

5. Отсутствие у функции одновременно стрелок управления и входа не допускается. Это означает, что запуск данной функции не контролируется и может произойти в любой произвольный момент времени либо вообще никогда.

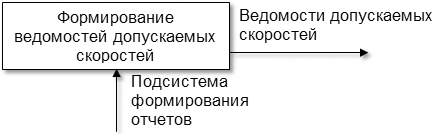


Рис. 6.14. Функция без управления и входа

Блок с наличием только управления можно рассматривать как вызов в программе функции (процедуры) без параметров. Если у блока имеется вход, то он эквивалентен вызову в программе функции с параметрами. Таким образом, блок без управления и входа эквивалентен функции, которая в программе ни разу не вызывается на исполнение.

На рис. 6.7–6.12, отображающих фрагменты диаграмм IDEF0, встречаются блоки без входа и управления. Это не стоит рассматривать как ошибку, так как подразумевается, что одна из этих стрелок должна быть.

6. У каждого блока должен быть как минимум один выход.

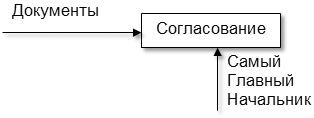


Рис. 6.15. Функция без выхода

Работы без результата не имеют смысла и не должны моделироваться. Исключение составляют работы, отображаемые в модели AS-IS. Их наличие свидетельствует о неэффективности и несовершенстве технологических процессов. В модели TO-BE эти работы должны отсутствовать.

7. При построении диаграмм следует минимизировать число пересечений, петель и поворотов стрелок.

8. Обратные связи и итерации (циклические действия) могут быть изображены с помощью обратных дуг. Обратные связи по входу рисуются «нижней» петлей, обратная связь по управлению – «верхней» (см. рис. 6.4 и 6.6).

9. Каждый блок и каждая стрелка на диаграммах должны обязательно иметь имя. Допускается использовать ветвление (декомпозицию) или слияние (композицию) стрелок. Это связано с тем, что одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. И наоборот, одинаковые или однородные данные и объекты, порожденные разными работами, могут использоваться в одном месте.

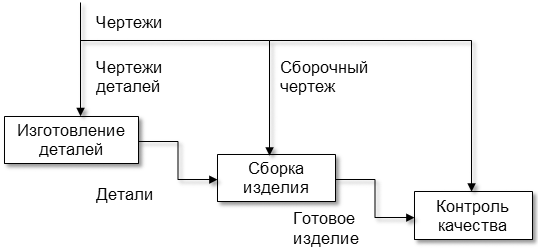


Рис. 6.16. Ветвление стрелок

При этом допускается задание различным ветвям стрелки уточняющих имен после разветвления (до слияния). Если какая-либо ветвь после ветвления не именована, то считается, что ее имя соответствует имени стрелки, записанному до ветвления.

Так, на рис. 6.16 управления, входящие в блоки «Изготовление деталей» и «Сборка изделия», имеют уточняющие значения и являются составной частью более общего управления «Чертежи». Для работы блока «Контроль качества» используются все чертежи.

На диаграмме не допускается рисовать стрелки, когда до и после ветвления они не именованы. На рис. 6.17 стрелка, входящая в блок «Формирование типовых ведомостей», не имеет имени до и после ветвления, что является ошибкой.

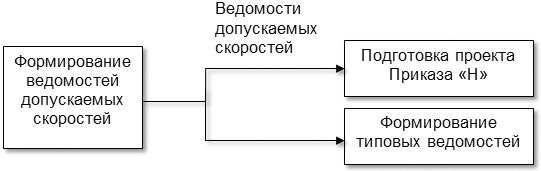


Рис. 6.17. Неправильное именование стрелок

10. При построении диаграмм для лучшей их читаемости может использоваться механизм туннелирования стрелок. Например, чтобы не загромождать лишними деталями диаграммы верхних уровней (родительские), на диаграммах декомпозиции начало дуги помещают в тоннель.

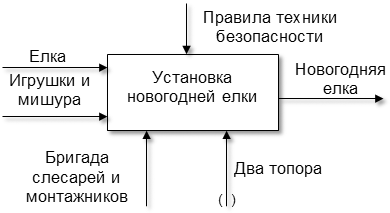


Рис. 6.18. Туннелирование стрелок

В данном примере при построении модели проведения новогоднего утренника механизм «два топора» не будет отображаться на диаграммах верхних уровней, при чтении которых может возникнуть справедливый вопрос: «А зачем нужны два топора на новогоднем утреннике?».

Аналогичным образом можно выполнять туннелирование с обратной целью – недопущения отображения стрелки на диаграммах низших уровней. В этом случае круглые скобки ставятся на конце стрелки. На контекстной диаграмме (см. рис. 6.21) затуннелирован механизм «Инженер службы пути», входящий в блок «Определение допускаемых скоростей». Такое решение принято, так как инженер непосредственно участвует во всех работах, отображенных на диаграмме декомпозиции этого блока (см. рис. 6.22). Чтобы не показывать эту связь и не загромождать диаграмму декомпозиции, стрелка была затуннелирована.

11. Все стрелки, входящие и выходящие из блока, при построении для него диаграммы декомпозиции должны быть отображены на ней. Исключение составляют затуннелированные стрелки. Имена стрелок, перенесенных на диаграмму декомпозиции, должны совпадать с именами, указанными на диаграмме верхнего уровня.

12. Если две стрелки проходят параллельно (начинаются из одной и той же грани одной работы и заканчиваются на одной и той же грани другой работы), то по возможности следует их объединить и называть единым термином.

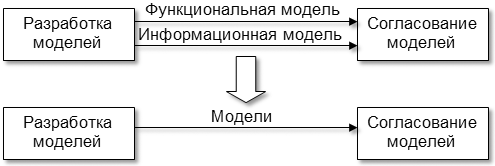


Рис. 6.19. Объединение связей

13. Каждый блок на диаграммах должен иметь свой номер. Для того чтобы указать положение любой диаграммы или блока в иерархии, используются номера диаграмм. Блок на диаграмме верхнего уровня обозначается 0, блоки на диаграммах второго уровня – цифрами от 1 до 9 (1, 2, …, 9), блоки на третьем уровне – двумя цифрами, первая из которых указывает на номер детализируемого блока с родительской диаграммы, а вторая номер блока по порядку на текущей диаграмме (11, 12, 25, 63) и т. д. Контекстная диаграмма имеет обозначение «А – 0», диаграмма декомпозиции первого уровня – «А0», диаграммы декомпозиции следующих уровней – состоят из буквы «А», за которой следует номер декомпозируемого блока (например, «А11», «А12», «А25», «А63»). На рисунке показано типичное дерево диаграмм (диаграмма дерева узлов) с нумерацией.

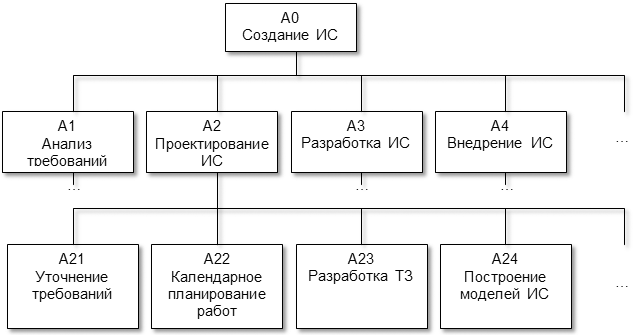


Рис. 6.20. Иерархия диаграмм

В современных CASE-средствах механизмы нумерации работ поддерживается автоматически. CASE-средства обеспечивают также автоматическое построение диаграмм дерева узлов, которые содержат только иерархические связи. Вершиной такой диаграммы может быть любой узел (блок), и она может быть построена на любую глубину.

**Пример построения модели IDEF0 для системы определения допускаемых скоростей**

Расчет допускаемых скоростей движения поездов является трудоемкой инженерной задачей. При проходе поездом какого-либо участка фактическая скорость движения поезда не должна превышать предельно допускаемую. Эта предельно допускаемая скорость устанавливается исходя из опыта эксплуатации и специально проводимых испытаний по динамике движения и воздействию на путь подвижного состава. Непревышение этой скорости гарантирует безопасность движения поездов, комфортабельные условия езды пассажиров и т. п. Они определяются в зависимости от типа подвижного состава (марки локомотива и типа вагонов), параметров верхнего строения пути (типа рельсов, балласта, эпюры шпал) и плана (радиуса кривых, переходных кривых, возвышения наружного рельса и т. д.). Как правило, для установления допускаемых скоростей необходимо определить не менее двух (на прямых) и пяти (в кривых) скоростей, из которых и выбирается окончательная допускаемая скорость, как наименьшая из всех рассчитанных. Расчет этих скоростей регламентируются Приказом МПС России № 41 от 12 ноября 2001 г. «Нормы допускаемых скоростей движения подвижного состава по железнодорожным путям колеи 1520 (1524) мм Федерального железнодорожного транспорта».

Как было отмечено, построение модели IDEF0 начинается с представления всей системы в виде простейшей компоненты (контекстной диаграммы). Данная диаграмма отображает назначение (основную функцию) системы и необходимые входные и выходные данные, управляющую и регламентирующую информацию, а также механизмы.

Контекстная диаграмма для задачи определения допускаемых скоростей показана на рис.6.21. Для построения модели использовался продукт BPwin 4.0 фирмы Computer Associates.

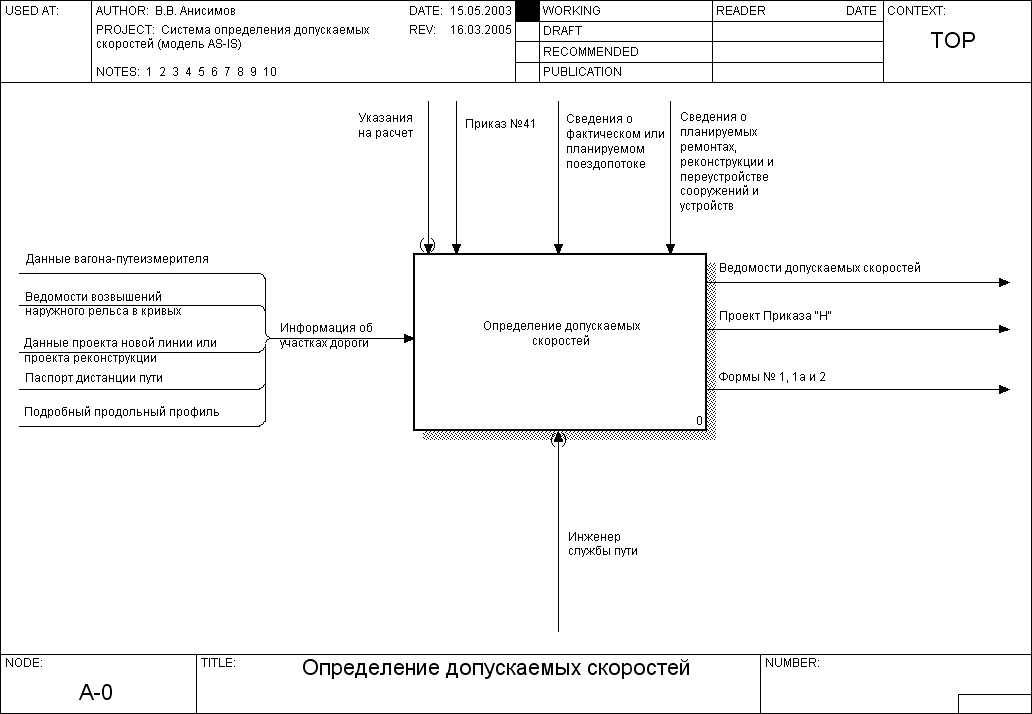


Рис. 6.21. Контекстная диаграмма системы определения допускаемых скоростей (методология IDEF0)

В качестве *исходной информации*, на основе которой выполняется определение допускаемых скоростей, используются:

- данные проекта новой линии или проекта реконструкции (содержат всю необходимую информацию для реализации проекта, а именно километраж, оси раздельных пунктов, план линии и др.);

- подробный продольный профиль (содержит информацию, аналогичную рассмотренной выше);

- паспорт дистанции пути (содержит информацию, аналогичную рассмотренной выше, а также сведения о верхнем строении пути (ВСП));

- данные о результатах съемки плана пути вагоном-путеизмерителем;

- ведомость возвышений наружного рельса в кривых (содержит информацию о плане пути).

Часть исходной информации может быть взята из разных источников. В частности, сведения о плане (параметрах кривых) могут быть взяты из проекта новой линии или проекта реконструкции, подробного продольного профиля, паспорта дистанции пути и т.д.

*Управляющими данными* являются:

- указание начальника службы пути дороги или Департамента пути и сооружений ОАО «РЖД» на расчет;

- Приказ № 41, содержащий нормативно-справочную информацию, порядок и формулы определения допускаемых скоростей;

- сведения о текущем или планируемом поездопотоке (данные о марках обращающихся локомотивов и типах используемых вагонов);

- сведения о планируемых ремонтах пути, реконструкции и переустройстве сооружений и устройств.

*Результатом* работы системы должны быть:

- ведомости допускаемых скоростей, содержащие все типы рассчитанных скоростей и позволяющие установить причину их ограничения;

- ведомости Приказа начальника дороги об установлении допускаемых скоростей на перегонах и раздельных пунктах (Приказ «Н») согласно принятой на дороге форме. Утвержденный Приказ «Н» официально закрепляет допускаемые скорости движения поездов;

- типовые формы № 1, 1а и 2, содержащие планируемые допускаемые скорости для разработки графика движения поездов.

Скорости, содержащиеся в Приказе «Н» и типовых формах, могут отличаться от рассчитанных и показываемых в ведомостях допускаемых скоростей. Это связано с тем, что в них отражают ограничения скорости не только по конструкции подвижного состава, параметров ВСП и кривых, но и по состоянию устройств и сооружений (деформация земляного полотна, перекос опор контактной сети и т. д.). Кроме того, они корректируются с учетом планируемых ремонтов пути, реконструкции и переустройства сооружений и устройств и т.д.

После построения контекстная диаграмма детализируется с помощью диаграммы декомпозиции первого уровня. На этой диаграмме отображаются функции системы, которые должны быть реализованы в рамках основной функции. Диаграмма, для которой выполнена декомпозиция, по отношению к детализирующим ее диаграммам называется ***родительской***. Диаграмма декомпозиции по отношению к родительской называется ***дочерней***.

Диаграмма декомпозиции первого уровня для рассматриваемой задачи приведена на рис.6.22. Как правило, при построении диаграммы декомпозиции исходная функция (декомпозируемая) разбивается на 3–8 подфункций (блоков). При этом блоки на диаграмме декомпозиции рекомендуется располагать слева направо сверху вниз, чтобы лучше была видна последовательность и логика взаимодействия подфункций.

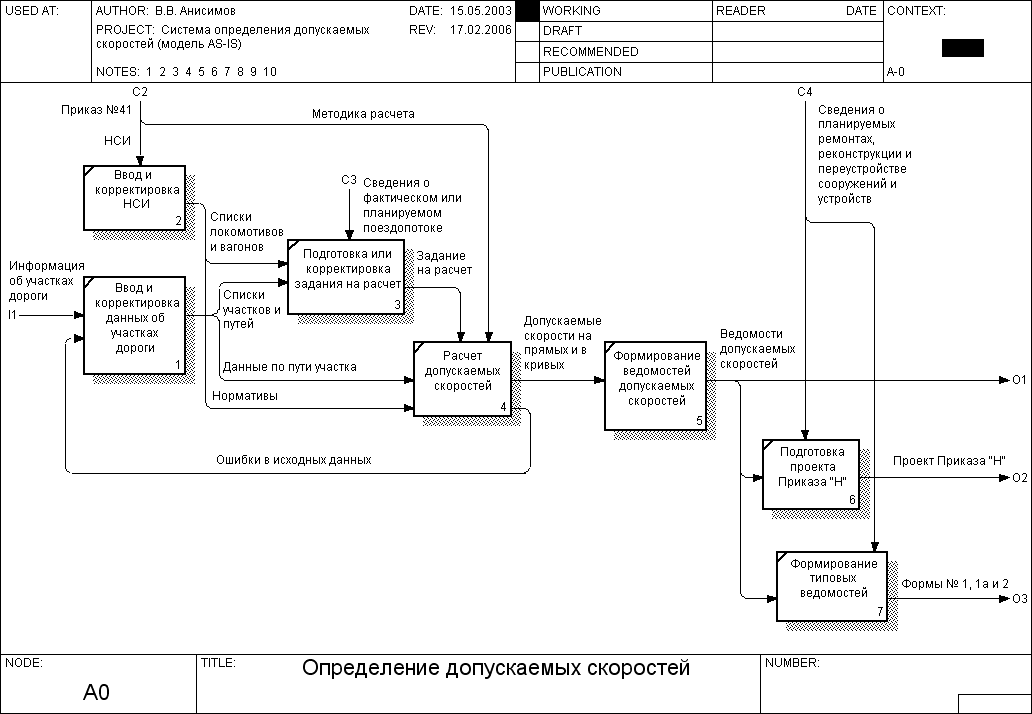


Рис. 6.22. Диаграмма декомпозиции первого уровня (методология IDEF0)

Очередность выполнения функций для решения рассматриваемой задачи следующая:

- ввод и корректировка нормативно-справочной информации и данных по участкам дороги (блоки 1 и 2);

- подготовка задания на расчет (блок 3). В нем указывается, для какого участка и пути, а также марки локомотива и типа вагонов следует выполнить расчет;

- расчет допускаемых скоростей в соответствии с порядком и формулами, указанными в Приказе № 41 (блок 4). В качестве исходной информации выступают данные по пути участка (план, верхнее строение пути и т. д.) и нормативы, выбираемые на основании задания на расчет;

- формирование ведомостей допускаемых скоростей (блок 5). На базе результатов расчета создаются несколько видов выходных документов, которые, с одной стороны, позволяют выявить причину ограничений скорости, с другой стороны, выступают в качестве основы для подготовки регламентированных документов;

- формирование и подготовка проекта Приказа «Н» и типовых ведомостей (блоки 6 и 7).

После построения диаграммы декомпозиции первого уровня для указанных на ней функций строятся отдельные диаграммы (диаграммы декомпозиции второго уровня). Затем процесс декомпозиции (построения диаграмм) продолжается до тех пор, пока дальнейшая детализация функций не теряет смысла. Для каждой атомарной функции, описывающей элементарную операцию (т. е. функции, не имеющей диаграмму декомпозиции), составляется подробная спецификация, определяющая ее особенности и алгоритм реализации. В качестве дополнения к спецификации могут использоваться блок-схемы алгоритмов. Таким образом, процесс функционального моделирования заключается в постепенном выстраивании иерархии функций.