МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНЫХ РАБОТ
по курсу
«ХРАНИЛИЩА ДАННЫХ»

Томск - 2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра автоматизации обработки информации (АОИ)

УТВЕРЖДАЮ
Зав. Кафедрой АОИ
Д.т.н., профессор
_____
Ю. П. Ехлаков
«___»
_____ 2015 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Для выполнения лабораторных работ
по дисциплине «Хранилища данных»
для студентов направления
230102 – Бизнес информатика

Разработчик
Доцент каф. АОИ
К.т.н., с.н.с.

О.И. Жуковский

Томск 2015

УДК 681.324:002.5

Аннотация

Целью данной работы является получение и развитие навыков в планировании **бизнес-процессов разработки хранилища данных** на основе функциональных моделей, отвечающих методологии IDEF0.

Рассматриваются основные бизнес-процессы разработки хранилища данных и базовые принципы функционального моделирования. Основное внимание уделено технологическому процессу создания моделей. Приводятся практические рекомендации по выполнению отдельных этапов процесса моделирования.

Для самостоятельного моделирования предлагаются варианты предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении.

Оглавление

Введение

В данном пособии рассматриваются основные положения планирования **бизнес-процессов разработки хранилища данных** и необходимые для этого основные понятия, методы и процедуры, связанные с построением функциональных моделей семантических характеристик данных с использованием методологии SADT (IDEF0). SADT – это аббревиатура слов Structured Analysis and Design Technique (технология структурного анализа и проектирования).

Семантические модели данных могут использоваться в сфере бизнеса для управления данными как ресурсами, интеграции информационных систем и построения систем управления как базами данных так и организованными на их основе системами хранилищ данных.

Необходимость в семантических моделях данных была впервые осознана ВВС США в середине семидесятых годов при выполнении программы ICAM (программы интегрированной компьютеризации производства). Цель этой программы состояла в повышении продуктивности производства посредством систематического внедрения компьютерных технологий. Повышение продуктивности производства в рамках программы ICAM потребовало совершенствования методов анализа и передачи информации. В результате был разработан ряд технологий, известных как IDEF-методологии (ICAM DEFinition).

1.Типовая модель бизнес-процессов разработки хранилища данных

Рассмотрим типовую модель бизнес-процессов разработки ХД, которая может быть положена в основу реализации любого конкретного проекта. Эта модель содержит минимально достаточное число обязательных этапов для реализации небольшого или среднего по масштабу проекта.

Проект разработки ХД начинается после того, как выбраны инструменты разработки и сформирована команда проекта. В проектный цикл разработки ХД обычно включаются следующие типовые процессы (этапы):

• формулирование требований;

• создание вычислительной среды;

• моделирование данных;

• определение процедур извлечения преобразования и загрузки данных;

• проектирование аналитических отчетов;

• разработка приложений ХД;

• настройка производительности;

• проверка качества;

• передача системы складирования данных в эксплуатацию. Опишем каждый типовой этап разработки ХД по следующей схеме:

- **Описание задачи.** Что обычно должно быть достигнуто в течение данного этапа разработки ХД.
- **Временные требования.** Приблизительная оценка количества времени для решения задачи данного этапа.
- **Результат выполнения этапа.** В конце каждого этапа оформляются один или более документов, которые описывают шаги и результаты решения данной задачи.
- **Потенциальные опасности.**

Теперь перейдем к рассмотрению отдельных компонентов представленной бизнес-модели.

1.1 Формулирование требований

Задача этапа. Главной задачей данного этапа является идентификация требований заказчика ХД и оформление их в виде документа «Каталог требований». Обычно сбор требований осуществляется путем опроса групп потенциальных пользователей ХД на специальных совещаниях и переговорах. Конечные пользователи, как правило, не знакомы с концепцией ХД и процессом складирования данных. Поэтому для успешного решения этой задачи важна помощь лиц, принимающих решения (ЛПР), т.е. руководителей организации. На этом этапе определяются:

- масштаб проекта создания ХД (границы предметной области);
- перечень и содержание отчетов;
- требования по анализу данных (перечень задач анализа данных);
- требования к аппаратному обеспечению;
- требования к системному программному обеспечению;
- значения базовых параметров Хранилища данных (скорость обработки запросов, объемы используемых данных, актуализация данных, производительность системы и т.д.);
- требования к квалификации персонала (программа обучения персонала);
- перечень источников данных;
- конкретизация плана реализации проекта разработки ХД (определяется дата завершения проекта).

В дополнение, основываясь на собранной выше информации, составляется план восстановления хранилища данных в случае аварийных сбоев. Разрабатывается стратегия архивирования и восстановления ХД.

Временные требования. Время выполнения этапа — от двух недель до двух месяцев.

Результатом выполнения этапа являются каталог требований, утвержденный заказчиком, и уточненный план проекта, который точно

определяет используемые ресурсы и даты контрольных точек проверки хода выполнения проекта.

Потенциальные опасности. Этап формулирования требований часто оказывается одним из самых узких мест проекта создания ХД. Причина состоит в конфликте внутрикорпоративных интересов и в необходимости наладить коммуникации для успешного выполнения и этапа и проекта в целом.

По определению складирование данных предполагает интеграцию в ХД данных из нескольких источников (подразделений организации). Позиция и взгляды подразделений на бизнес-информацию (производство данных, их верификацию, поставку данных в ХД, разграничение доступа к данным и т.д.) могут сильно расходиться и даже быть диаметрально противоположными. Даже если команда разработчиков предложит блестящее решение по созданию ХД, его реализация может споткнуться о нежелание определенных групп потенциальных пользователей предоставлять данные в ХД или конструктивно участвовать в определении требований к системе складирования данных.

Если не удается наладить коммуникации между участниками процесса, то все усилия по созданию ХД будут потрачены впустую и проект никогда не будет завершен в установленные сроки, а в худшем случае просто будет провален.

Одним из способов избежать такой ловушки является непосредственное вовлечение руководителей организации в процесс реализации проекта. Следует заручиться поддержкой влиятельного покровителя из числа высшего руководства, чтобы можно было влиять на позицию подразделений сотрудничать с командой разработчиков ХД.

1.2 Создание вычислительной среды для хранилища данных

Задача этапа. Главной задачей этого этапа является создание информационно-вычислительной среды, в которой будет разрабатываться ХД. Это достаточно важный этап. Совершенно ясно, что создание ХД организации на двух компьютерах разработчиков — затея весьма опасная и нереальная.

Вычислительная среда разработки ХД данных должна достаточно точно моделировать ту вычислительную среду, в которой реально будет эксплуатироваться ХД, т.е. она должна быть масштабируемой по аппаратному решению.

После того как требования к программно-аппаратной части определены, нужно установить серверы приложений, серверы БД, клиентские рабочие станции и автоматизированные места разработчиков.

На этом этапе должна быть достигнута полная ясность в том, как будут выполняться технологические работы по созданию ХД. Из-за особенностей работы конкретной организации возможно применение нескольких вычислительных сред. Например, для кредитных организаций, которые работают в режиме реального времени, может быть использовано три различных вычислительных среды: программно-аппаратный комплекс разработчиков, программно-аппаратный комплекс для тестирования и вычислительная среда организации, в которой будет эксплуатироваться ХД.

Причина, по которой лучше применять отдельную вычислительную среду для разработки, состоит в том, что внесение изменений и тестирование разрабатываемого ХД могут привести к аварийным сбоям в существующей вычислительной среде организации, что повлечет дополнительные эксплуатационные издержки.

Временные требования.

Закупка и установка серверов — до двух недель.

Монтажные работы на установке компьютерной сети — две-четыре недели.

Результатом выполнения этапа являются спецификации на программно-аппаратное обеспечение, а также скрипты и установки для программного обеспечения.

Потенциальные опасности. Самой большой опасностью является использование одного сервера БД для моделирования различных

вычислительных сред, например, вычислительной среды разработки и вычислительной среды тестирования, или, что еще хуже, для вычислительной среды разработки и вычислительной среды эксплуатации ХД, особенно если на этом сервере работает уже существующая информационная система.

В этом случае могут возникнуть конфликты между различными участниками проекта создания ХД и ИТ-службой организации в случае непредусмотренной остановки сервера БД.

Кроме того, в тестовой среде будет невозможно промоделировать ожидаемые показатели нагрузки на систему и оценить ее производительность.

1.3 Моделирование данных

Задача этапа. Главной задачей этапа является разработка логической и физической моделей данных для ХД. Этот этап — один из самых важных для проекта создания ХД. Ошибки и просчеты, допущенные на этом этапе, будет очень сложно исправить на последующих этапах. Кроме того, подобные просчеты могут привести к пересмотру всего проекта и, следовательно, к его фактическому провалу.

При выполнении этого этапа сначала строится логическая модель данных, которая впоследствии преобразуется в физическую модель.

Одной из подзадач этого этапа являются идентификация и описание источников данных, которые также могут стать подзадачей этапа определения процедур извлечения, преобразования и загрузки данных в ХД (ETL-процессов).

Временные требования. Время выполнения этого этапа — от двух недель до двух месяцев.

Результатом выполнения этапа являются перечень источников данных и их описание, а также логическая и физическая модели данных.

Потенциальные опасности. Самой большой опасностью при выполнении этого этапа является самоуверенность проектировщиков ХД. Во многих случаях даже опытные проектировщики допускают от двух до пяти

ошибок в структуре данных (потерь существенных семантических зависимостей в данных) на проект. Причиной таких ошибок часто становится недостаточная осведомленность проектировщиков о предметной области ХД и низкое качество информации, поставляемой аналитиками предметной области.

Бизнес-аналитики предметной области могут не предоставить полной информации о функциональных зависимостях в данных, что приведет в результате к проектированию частично неправильной схемы данных. В этом случае схема правится «на лету» на последующих этапах выполнения проекта, что может привести к пересмотру архитектуры приложений и процессов ETL. Иногда ошибки носят принципиальный характер и приводят к полному пересмотру фрагмента схемы данных. Например, как в случае, когда пропущена зависимость вложения на подмножествах в данных (типа «часть-целое»).

Хорошей предупредительной мерой для предотвращения подобных ситуаций является привлечение квалифицированных экспертов, особенно в случае если проект разработки ХД выполняется силами самой организации.

1.4 Определение процедур извлечения, преобразования и загрузки данных

Задача этапа. Главной задачей этапа является идентификация и определение процедур извлечения, очистки (фильтрации), преобразования и загрузки данных. Это весьма трудоемкий этап, не столько по временным затратам, сколько по усилиям, предпринимаемым по анализу структур данных источников и подающих систем.

Исключительно редким является случай, когда ХД данных создается на голом месте, т.е. в подразделениях отсутствуют автоматизированные подсистемы обработки данных. Как правило, данные уже существуют (в том или ином виде). Их нужно собрать, согласовать, привести к единому формату, агрегировать и загрузить в ХД. По этой причине этот этап является характерным для проекта создания ХД.

Также следует помнить о том, что сам процесс подготовки и загрузки данных в создаваемое ХД может занять более половины времени, отведенного на реализацию проекта, особенно в проектах большого масштаба.

Временные требования. Время выполнения этапа — от одной недели до полутора месяцев.

Результатом выполнения этапа являются схема соответствия данных подающих систем и ХД, программы или ETL-инструменты.

Потенциальные опасности. Первой потенциальной опасностью при выполнении этого этапа является недооценка временных параметров. Обычно на выполнение этапа выделяют немного времени. Процедуры извлечения, преобразования и загрузки данных в ХД оказывают непосредственное влияние на качество и полноту предоставляемой информации конечным пользователям. Недостаточное внимание к разработке этих процедур может вызвать негативные реакции у пользователей после получения ими отчетов.

Второй потенциальной опасностью является стремление команды разработчиков сделать процесс ETL как можно более всеобъемлющим, мотивируя свои действия стремлением обеспечить качество данных. Следует помнить, что главная цель ETL-процесса — оптимизации скорости загрузки данных в ХД.

1.5 Проектирование аналитических отчетов

Задача этапа. Главной задачей выполнения этого этапа является проектирование и разработка аналитических отчетов на спроектированной структуре данных. Это также специфичный этап для проекта ХД.

Перечень требуемых отчетов содержится в «Каталоге требований», и их разработка решает в целом поставленную задачу. Однако следует помнить, что потенциальные пользователи не всегда точно знают, что они хотят увидеть в отчетах. С другой стороны, как правило, собранные данные предоставляют большие возможности по формированию разнообразных отчетов, чем это

зафиксировано в «Каталоге требований». Здесь должен быть найден разумный компромисс.

Временные требования. Время выполнения этого этапа зависит от числа разрабатываемых отчетов. В зависимости от сложности отчета его разработка занимает от 4 часов до двух недель.

Результатом выполнения этапа станут спецификация кубов данных (измерения и метрики) и разработанные отчеты.

Потенциальные опасности. Потенциальной опасностью при выполнении этого этапа является то, что не уделяется достаточного внимания оптимизации времени получения отчета. Конечный пользователь не любит долго ждать. А если он получал такой отчет на своей настольной системе быстрее, то мнение о ХД у него будет, мягко говоря, неадекватное.

Хороший способ избежать такой опасности — тестирование разработанных отчетов с целью минимизации времени их получения.

1.6 Разработка приложений хранилища данных

Задача этапа. Главной задачей данного этапа является формирование программной среды, в которой пользователи будут извлекать данные из ХД и просматривать предопределенные отчеты.

Каковы бы ни были инструментальные средства OLAP, необходимо продумать, как будут происходить визуализация отчетов и их доставка конечному пользователю.

Временные требования. Срок выполнения этого этапа — от одной недели до месяца. При использовании тонкого клиента типа браузера следует исходить из того, что в среднем на разработку и отладку одной веб-страницы тратится полдня.

Результатом выполнения этапа будет документация, описывающая механизм доставки пользователям отчетов и спецификации экранных форм.

Потенциальные опасности. Самой большой потенциальной опасностью является ложное представление о достаточной квалификации пользователей ХД

для работы с ИТ- технологиями. Конечные пользователи хотят быстро получать данные, необходимые для решения своих конкретных задач, а не изучать многотомные инструкции по работе с программным обеспечением.

Если интерфейс интуитивно не понятен или не содержит ясных и кратких инструкций по работе с электронной формой (встроенных в форму Help), он не будет использовать систему складирования данных с должной эффективностью, что приведет к постепенному ее вымиранию.

1.7 Настройка производительности

Задача этапа. Главная задача выполнения этого этапа — добиться оптимальной производительности ETL-процессов, производства отчетов и их доставки конечному пользователю.

Извлечение, преобразование и загрузка данных, как правило, является длительным процессом, который выполняется в пакетном режиме с не очень высоким приоритетом. Пока данные не загружены в ХД, новые отчеты не могут быть получены.

При обработке некоторых запросов захватывается большое количество данных, на которых выполняются динамические операции агрегирования и суммирования, что приводит к значительному времени производства отчета.

На доставку отчетов пользователю может влиять загрузка локальной вычислительной сети и большой объем сетевого трафика.

Временные требования. Время выполнения этого этапа не должно превышать одну-две недели.

Результатом выполнения этапа является перечень рекомендаций по настройке производительности.

Потенциальные опасности. Потенциальной опасностью этого этапа считается использование вычислительной среды разработки ХД, которая не масштабируется к вычислительной среде эксплуатации ХД. Из-за этого рекомендации по настройке производительности не будут соответствовать реальным условиям эксплуатации ХД.

Проверка качества

Задача этапа. Главной задачей этапа — убедиться, что ХД готово к эксплуатации. Как правило, проверка качества выполняется отдельной группой специалистов, не входящих в состав команды разработчиков.

Временные требования. Срок выполнения этого этапа — от одной до четырех недель.

Результатом выполнения этапа являются план тестирования ХД и заключение о готовности ХД к эксплуатации.

Потенциальные опасности. Самый большой риск любого проекта — это люди: их квалификация, амбиции, заинтересованность в деле, мотивы и т.д. Поскольку люди, проверяющие ХД на этом этапе, не входят в проектную команду, возникает потенциальная опасность, связанная с недостаточной образованностью этих людей в области складирования данных. Разумным решением при выполнении этого этапа является привлечение организацией сторонних специалистов высокой квалификации по проверке качества ХД.

1.8 Передача системы ведения хранилища данных в эксплуатацию

Задача этапа. Главная задача этапа — передача системы ведения хранилища данных заказчику и представление ее конечным пользователям.

Временные требования. Срок выполнения этого этапа — от одного дня до нескольких недель.

Результатом выполнения этапа является акт о приемке-сдаче программного продукта.

Потенциальные опасности. Самая большая потенциальная опасность для этого этапа — неготовность потенциальных пользователей к работе с ХД. Хорошим правилом здесь будет проведение нескольких презентаций и коротких обучающих курсов по работе с ХД.

1.9 Сопровождение и модификация хранилища данных

Обычно в проектный цикл включают еще два этапа — этап сопровождения ХД и этап его модификации. Это важные этапы в жизни ХД. Однако, как показывает опыт, целесообразно выделять их в самостоятельные проекты.

Было бы большой ошибкой поручать разработчикам ХД его сопровождение. Процессы сопровождения ХД требуют от ИТ-специалистов иной квалификации, чем процессы его разработки. Чтобы сопровождать, не обязательно уметь писать программы, но обязательно нужно уметь их настраивать и использовать.

Если необходимость в модернизации ХД возникает спустя несколько месяцев после сдачи его в эксплуатацию, это говорит о том, что проект не был успешным. Потребность в модернизации реально может сформироваться спустя шесть месяцев после интенсивной его эксплуатации, когда проверены его возможности, прочувствована отдача и видны новые перспективы использования, т.е. когда ХД стало частью производственного процесса. При этом не факт, что та же команда разработчиков будет заниматься его модернизацией.

Резюме

Процесс разработки ХД может быть представлен в виде модели бизнес-процессов. Бизнес-модель процесса разработки позволяет:

отобразить субъективное мнение руководители ИТ и некоторых участников команды на процесс проектирования конкретного ХД;

- учесть особенности ИТ-проекта, в рамках которого проектируется ХД;
- достаточно быстро составить план проектирования конкретного ХД;
- просчитать длительность проектных работ (создать временную модель проектирования).

Представленная в настоящем пособии укрупненная бизнес-модель процесса разработки ХД является достаточно общей. В нее не включены многие детали и риски. Конкретизация и детализация процесса разработки ХД — это основная задача руководителя проекта разработки ХД. Проектировщик ХД должен понимать, какие документы он вправе ожидать от членов команды, какие документы должен создать и кому разработанные документы должен передать.

2. Создание функциональных моделей и диаграмм

2.1 Принципы функционального моделирования

Рассмотрим основные положения функционального моделирования систем. Под словом «система» будем понимать совокупность взаимодействующих компонент и взаимосвязей между ними. Мир, в котором мы живем, можно рассматривать как сложную взаимосвязанную совокупность естественных и искусственных систем. Это могут быть достаточно сложные системы (например, планеты в составе Солнечной системы), системы средней сложности (космический корабль) или сверхсложные системы (системы молекулярных взаимодействий в живых организмах). Существует огромное количество научных дисциплин, предназначенных для изучения и объяснения различных аспектов этого бесконечного спектра сложности. Например, механика может объяснить гравитационное притяжение двух планет, а химия может описать молекулярные взаимодействия в стакане кипятка. Искусственные системы по своей сложности, как правило, занимают среднее положение. Например, всемирная телефонная сеть содержит десятки или даже сотни тысяч переключателей, однако количество взаимодействий этих переключателей не идет ни в какое сравнение с количеством взаимодействий молекул даже в небольшом стакане воды. С точки зрения общей теории систем такие системы обычно рассматриваются как системы средней сложности.

Под термином «моделирование» будем понимать процесс создания точного описания системы. Особенно трудным оказывается описание систем средней сложности, таких, как система коммутаций в телефонных сетях, управление аэровоздушными перевозками или движением подводной лодки, сборка автомобилей, челночные космические рейсы, функционирование перерабатывающих предприятий. С точки зрения человека, эти системы описать достаточно трудно, потому что они настолько велики, что практически невозможно перечислить все их компоненты со своими взаимосвязями, и в то же время недостаточно велики для применения общих упрощающих

предположений (как это принято в физике). Наша неспособность дать простое описание, следовательно, и обеспечить понимание таких систем делает их проектирование и создание трудоемким и дорогостоящим процессом и повышает степень их ненадежности. С ростом технического прогресса адекватное описание систем становится все более актуальной проблемой.

Описание системы с помощью SADT называется моделью. В SADT-моделях используются как естественный, так и графический языки. Для передачи информации о конкретной системе источником естественного языка служат люди, описывающие систему, а источником графического языка – сама методология SADT. В дальнейшем вы увидите, что графический язык SADT обеспечивает структуру и точную семантику естественному языку модели. Графический язык SADT организует естественный язык вполне определенным и однозначным образом, за счет чего SADT и позволяет описывать системы, которые до недавнего времени не поддавались адекватному представлению.

Одна и та же схема моделирования может быть использована для моделирования любого выбранного объекта. Универсальной единицей пунктуации для неограниченного строго структурного анализа является SA-блок (рис. 2.1):

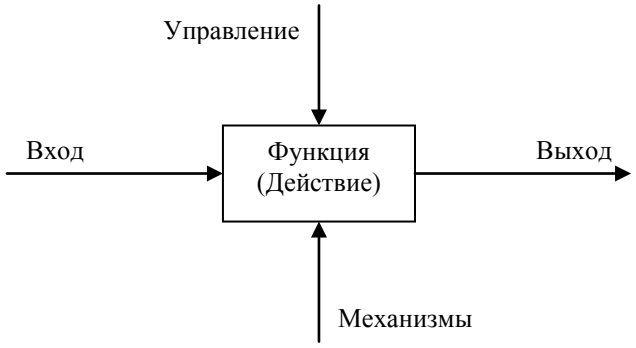


Рис. 1.1. Пример SA-блока

Вход при наличии **управления** преобразуется в **выход** с помощью **механизма.**

Выходы одного блока могут быть входами или управлениями (или исполнителями) для других блоков. (Блоки именуются, а дуги помечаются с

использованием естественного языка.) Дуги могут разветвляться и соединяться, а каждый блок может быть подвергнут декомпозиции, т. е. разделен как целое на свои составляющие на более детальной диаграмме. Входы, управления и выходы определяют интерфейсы между блоками, а исполнители позволяют при необходимости в определенной степени объединять объекты. Границы блоков и диаграмм должны быть согласованы, а возникающая иерархическая, взаимосвязанная совокупность диаграмм является моделью.

Диаграмма ограничивается 3-6 блоками для того, чтобы детализация осуществлялась постепенно. Вместо одной громоздкой модели используется несколько небольших взаимосвязанных моделей, значения которых взаимно дополняют друг друга, делая понятной структуризацию сложного объекта.

2.2. Сбор информации

Опрос – это сбор сведений. Первый опрос служит точкой отсчета в процессе моделирования. Чтобы провести опрос, аналитик вначале выбирает наилучший источник информации (документ или конкретного человека), а затем организует его «опрос». Цель опроса – получение порции информации, необходимой для начала либо для продолжения построения определенной части модели. После первого опроса модель используется для определения той информации, которую необходимо получить в ходе следующего опроса. В соответствии с иерархией модели может быть проведена последовательность опросов для выяснения все более конкретных деталей рассматриваемой области.

SADT-аналитики используют свои модели для более сфокусированного опроса и оптимизации затрат времени на работу с источниками информации. Это повышает эффективность работы, сокращает время на повторное рассмотрение неясных или забытых деталей и уменьшает вероятность повторно задавать вопросы одному и тому же эксперту. При таких условиях эксперты более охотно отвечают на вопросы аналитиков и тем самым спасают

аналитические проекты от одного из «убийц» – прекращения потока информации.

Обычно источниками информации служат эксперты. Часто именно они являются наилучшими источниками, потому что им знакомы текущие нюансы и недокументированные аспекты системы. Самое важное – это то, что экспертам известны факты, которые не отражены в документах или которые трудно объяснить. Их можно получить только путем опроса экспертов. Чтобы подготовиться к такому опросу, следует исследовать другие источники информации, например документы. Существует множество различных стратегий для извлечения информации из этих источников. Вот те, которые обычно принято использовать:

- чтение документов;
- наблюдение за выполняемыми операциями;
- анкетирование;
- использование собственных знаний;
- составление описания.

Документы – хороший источник информации, потому что они чаще всего доступны и их можно «опрашивать» в удобном для себя темпе. Чтение документов – прекрасный способ получить первоначальное представление о системе и сформулировать вопросы к экспертам. Для SADT-проектов такие документы могут храниться в библиотеке проекта и распространяться в виде небольших рабочих пакетов, называемых папками.

Наблюдение за работой моделируемой системы – хорошая стратегия получения информации. Оно должно проводиться всегда, когда есть такая возможность. Через наблюдение, а возможно, и участие аналитики получают информацию о происходящих день за днем операциях из первых рук. Во время наблюдения за работой системы часто возникают вопросы, которые никогда бы не появились, если бы аналитик только читал документы или разговаривал с экспертами.

Анкетирование проводится для того, чтобы опросить большие группы экспертов в сжатые сроки. Его можно использовать, например, когда необходимо быстро получить сведения о работе какой-либо определенной части системы с разных позиций. Анкетирование при опросе экспертов позволяет выявить, какие части системы более всего нуждаются в улучшении.

Иногда опытные аналитики исследуют большое число систем определенного типа (например, производство, телефонная сеть, бухгалтерия). В ходе многих проектов они приобретают фундаментальные знания в соответствующей предметной области, относительно определенного класса систем. Такие знания очень полезны, поскольку для многих различных систем существуют общие понятия. Эти понятия могут иметь широкое применение. Если вы хорошо знаете предметную область, то можете стать сами источником информации.

Еще одна полезная стратегия – придумать описание и дать его экспертам для корректировки. Придуманные описания могут дать альтернативные схемы функционирования системы – схемы, о которых эксперты никогда не думали.

В процессе анализа, независимо от источников информации, проводятся опросы нескольких типов. Выбор того или иного типа зависит от вида необходимой информации и поставленной цели. Наиболее распространены следующие типы опросов:

- опросы для сбора фактов;
- опросы для определения проблем;
- совещания для принятия решений;
- диалоги автор/читатель.

Опросы для сбора фактов проводятся, когда пытаются определить, как функционирует система в настоящее время. Опросы для определения проблем полезны, когда вы хотите выяснить, что в системе не в порядке. Совещания для принятия решений проводятся, когда нужно получить представление о том, как должна функционировать будущая система, чтобы устранить недостатки в

настоящей. Диалоги автор/читатель – это неформальные обсуждения при разногласиях между автором и экспертом.

2.3. Начало моделирования

Начало моделирования в SADT означает создание диаграмм A0 и A-0, которые затем могут быть отрецензированы. Эти две диаграммы полностью рассказывают все об изучаемом процессе или системе с минимальной степенью детализации. Создавая их, аналитик предпринимает начальную попытку декомпозировать систему и затем обобщить полученную декомпозицию. Декомпозиция (диаграмма A0) освещает наиболее важные функции и объекты системы. Объединение (диаграмма A-0) трактует систему как «черный ящик», дает ей название и определяет наиболее важные входы, управления, выходы и, возможно, механизмы.

Прежде чем начать моделирование, SADT-аналитик проводит подготовку к нему, собирает информацию, декомпозирует объект и обобщает эту декомпозицию. Подготовка включает выбор цели модели (например, разработка нового программного продукта), выбор точки зрения, с которой будет представлена модель (например, группа разработки), тип создаваемой модели (например, модель «потокового» процесса) и предполагаемое использование построенной и проверенной модели (например, организация процесса разработки нового программного продукта). Таким образом, подготовка должна максимально облегчить сбор информации.

Сбор информации может включать любую комбинацию следующих видов деятельности: чтение документов, наблюдение за существующими операциями, анкетирование группы экспертов, опрос одного или нескольких экспертов, использование собственных знаний и придуманного описания работы системы, которое впоследствии может быть откорректировано.

Декомпозируя объект нужно, прежде всего, обратить внимание на входные и выходные данные для всей системы. Декомпозиция всей системы

начинается с составления списка основных типов данных и основных функций. Делая это, вы мысленно окидываете взглядом основные функции системы, рассматривая все нормальные и аномальные ситуации, обратные связи и случаи потенциальных ошибок. Потом эти списки снабжаются комментариями для указания основных типов, как данных, так и функций системы или их различных сочетаний. Наконец, списки с комментариями используются для создания диаграммы А0, которая затем обобщается с помощью диаграммы А-0.

Цель и точка зрения модели определяются на самой ранней стадии создания модели. Выбор цели осуществляется с учетом вопросов, на которые должна ответить модель, а выбор точки зрения – в соответствии с выбором позиции, с которой описывается система. Иногда цель и точку зрения можно выбрать до того, как будет сделана первая диаграмма. Например, цель модели разработки нового программного продукта можно определить заранее, потому что она очевидна в постановке задачи: «понять обязанности всех задействованных лиц так, чтобы организовать процесс разработки программного обеспечения». Настоятельно рекомендуется, как можно раньше определять цель и выбирать точку зрения новой модели. Но вначале попробуйте записать ряд специфических вопросов, на которые модель должна ответить, чтобы убедиться, что цель сформулирована точно, и рассмотрите систему с нескольких различных точек зрения, прежде чем выбрать одну из них. На рис. 2.1 показано, как это делается для задачи, связанной с разработкой нового программного обеспечения.

Иногда оказывается, что определить цель и точку зрения в самом начале моделирования чрезвычайно трудно. В таком случае следует составить списки данных и функций и, может быть, нарисовать диаграмму А0. Сделав это, вы начнете чувствовать систему и установите, описывает ли ее диаграмма А0 с нужной точки зрения. Может быть, вам придется нарисовать несколько альтернативных А0-диаграмм, прежде чем появится достаточная уверенность для того, чтобы осуществить выбор правильной цели и точки зрения.

Списки объектов системы, создаваемые в ходе моделирования, в SADT принято называть «списками данных». Термин «данное» здесь употребляется как синоним слова «объект». Следовательно, при обсуждении различных аспектов моделирования в SADT будем применять термин «список данных». Составление списка данных является начальным этапом создания каждой диаграммы функциональной SADT-модели. Правило заключается в том, чтобы вначале составить список данных, а потом список функций.

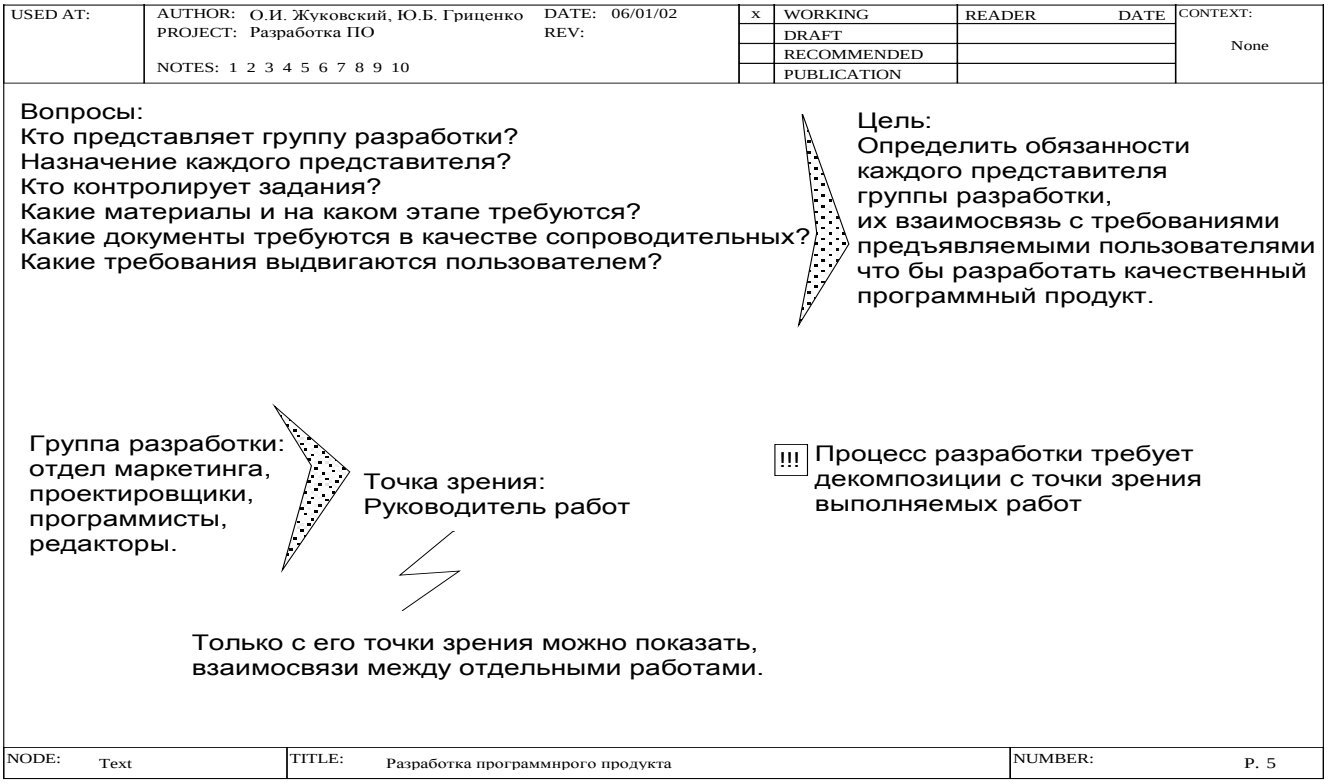


Рис. 2.1. Определение целей и точки зрения

USED AT:	AUTHOR: О.И. Жуковский, Ю.Б. Гриценко DATE: 06/01/02 PROJECT: Разработка ПО REV: NOTES: 1 2 3 4 5 6 7 8 9 10	x	WORKING	READER DATE	CONTEXT:
			DRAFT		None
			RECOMMENDED		
			PUBLICATION		

Функции:
планирование, разработка, классификация, управление, проектирование, измерение, контроль, изготовление, редактирование, тестирование, производство, учет, продажа, реализация, сбор

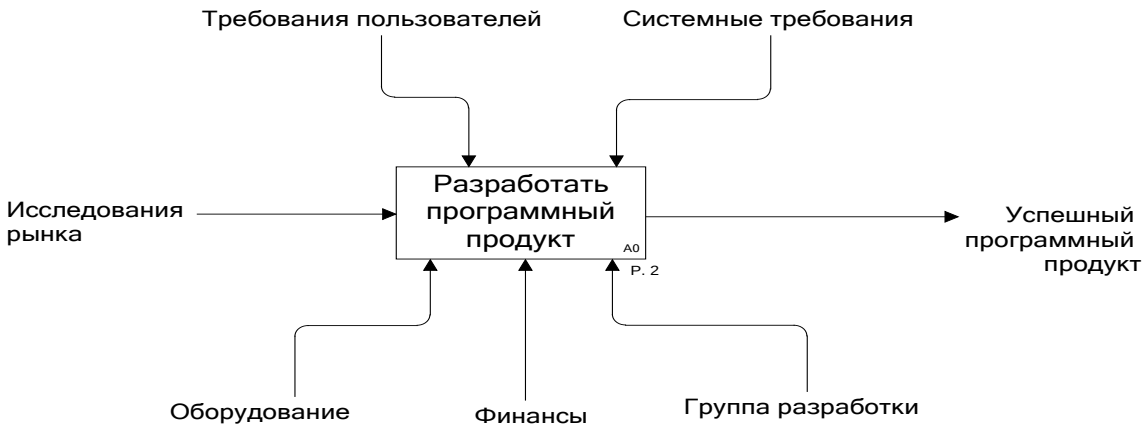
Управление:
инструкции, требования, чертеж, стандарты, запросы, правило, указания, заявки, задания, план, формат текста

Механизм:
персонал, компьютер, информационная система, фирма, станок, программное обеспечение, заказчик, аппаратура, подрядчик, инструмент, оператор, оснастка, проетировщик, руководитель проекта, программисты, тестировщики, редакторы

NODE: Text	TITLE: Разработка программнрого продукта	NUMBER: P. 6

Рис. 2.2. Подготовка списков функций и данных

USED AT:	AUTHOR: О.И. Жуковский, Ю.Б. Гриценко DATE: 10/01/02 PROJECT: Разработка ПО REV: NOTES: 1 2 3 4 5 6 7 8 9 10	x	WORKING	READER DATE	CONTEXT:
			DRAFT		Top
			RECOMMENDED		
			PUBLICATION		



Цель: Организовать процесс разработки программного обеспечения
Точка зрения: Руководитель разработки

NODE: A-0	TITLE: Разработка программнрого продукта	NUMBER: P. 1

Рис. 2.3. Диаграмма A-0

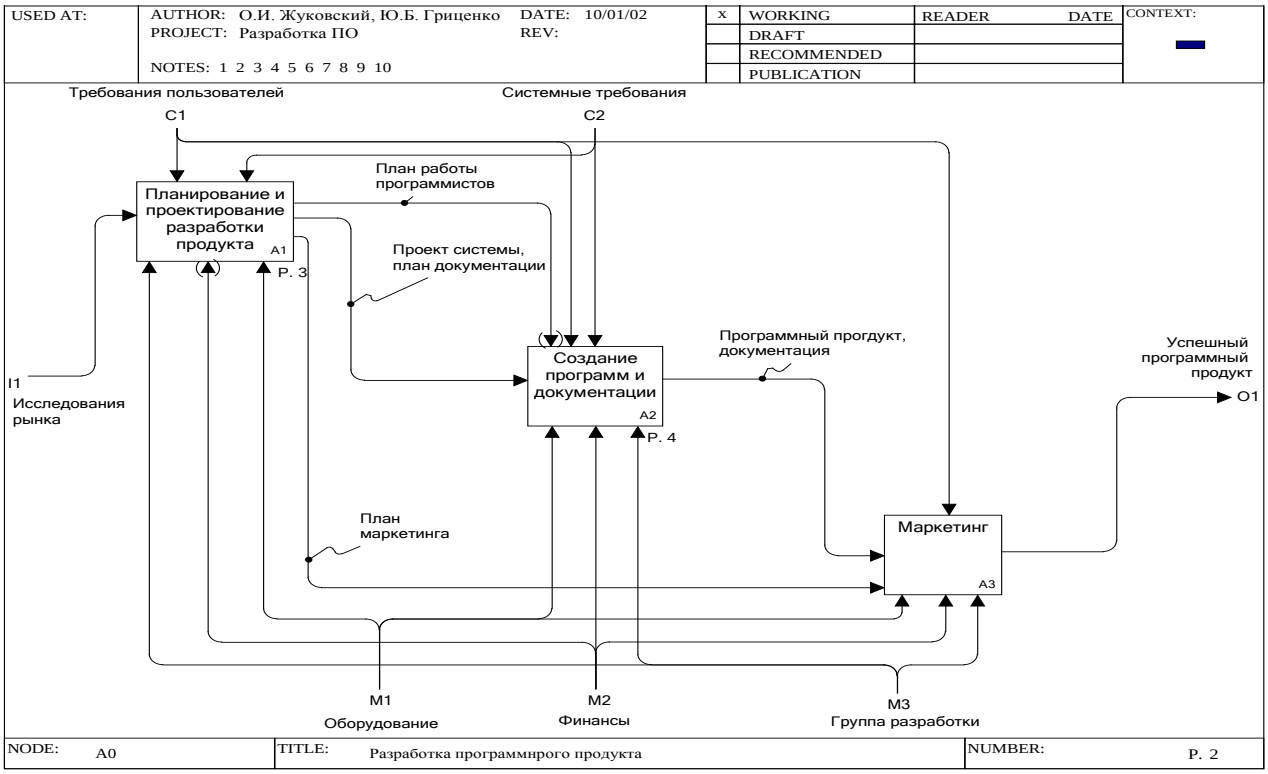


Рис. 2.4. Диаграмма A0.

Начните свою диаграмму с выделения всех основных групп и категорий данных, используемых и генерируемых системой. И не будьте слишком дотошны – запишите все разумные возможности. При сомнении записывайте все, что приходит на ум, потому что лучше записать слишком много, чем провести неполный анализ. Обратите внимание, что на рис. 2.2 в список вошло много деталей, хотя аналитик пытался создать диаграмму цеха как единого целого.

В современных аналитических методах слишком часто уделяется повышенное внимание функциям в ущерб данным. Начиная с составления списка данных, вы сможете избежать перехода к немедленной функциональной декомпозиции. Списки данных помогут выполнить более глубокий анализ и при этом не концентрироваться на функциях системы, избегая пробелов, которые часто возникают из предвзятых представлений о функциональных декомпозициях. Кроме того, вы сможете уделить должное внимание данным и

идентифицировать ограничения, определяющие функциональную декомпозицию.

SADT-диаграммы представляют границы функций и ограничения, накладываемые на них, причем ограничения должны присутствовать во всех системах. Указывая вначале ограничения, выявляем естественную структуру системы. Без ограничений функциональная SADT-диаграмма представляет собой не более чем схему потоков данных. Без ограничительных дуг диаграммы не смогут рассказать читателю, почему аналитик выбрал именно данную декомпозицию. Благодаря тому, что в SADT различаются входные дуги и дуги управления (информация, необходимая для пояснения декомпозиции), SADT-диаграммы ясно объясняют изучаемую систему и причину такой декомпозиции.

Закончив список данных, приступайте с его помощью к составлению **списка функций**. Для этого представьте себе функции системы, использующие тот или иной класс (тип) или набор данных. Помните, что несколько различных типов данных может использоваться одной функцией. Обозначьте, какие типы или наборы данных необходимы для каждой конкретной функции. Это позволит выделить данные сходных типов, которые затем можно объединить в метатипы.

По мере продвижения по списку, проверяйте, верны ли ваши первоначальные представления, которые часто могут не совпадать с выбранной целью и точкой зрения модели. С другой стороны, не следует автоматически отвергать первоначальные идеи, если они кажутся неверными. Дальнейшие размышления могут прояснить внутренние аспекты работы системы, не очевидные при первом взгляде, и вы, возможно, восстановите исходные идеи после построения нескольких других диаграмм.

Список функций должен находиться на одной странице со списком данных. При составлении исходного списка не пытаться объединять функции между собой. Вместо этого постарайтесь вначале сосредоточиться на каждой

конкретной функции и ее отношении к группам данных. Кроме того, старайтесь подбирать такие функции, которые могли бы работать с наиболее общими типами данных из вашего списка. Не будьте слишком дотошны, записывая функции, даже если вы сомневаетесь, выполняет ли их система. Что касается пограничных функций (функций, которые могут выполняться либо системой, либо ее окружением), то вначале очень трудно определить, входят они в модель или нет.

Затем объединяйте функции в «агрегаты». Стремитесь к организации 3-6 функциональных группировок. Старайтесь, чтобы эти группировки имели один и тот же уровень сложности, содержали примерно одинаковый «объем» функциональности, и функции в каждой из них имели сходные операции и цели. На рис. 2.2 видно, что исходный список функций сгруппирован в три функции более высокого уровня. Объединение не всегда легко осуществить. Вы можете обнаружить, что на каком-то уровне модели трудно выбрать «наилучший» способ объединения функций. Не волнуйтесь, потому что плохая группировка обнаружит свою слабость на этапе декомпозиции. Если это произойдет, вы всегда можете вернуться назад и попробовать другой вариант объединения.

Исходное содержание **диаграммы A0** обеспечивают списки данных и функций. Для правильного описания системы содержанию надо придать форму. В SADT это делается посредством построения диаграммы. Начинающим авторам необходимо придерживаться определенного порядка: (1) расположите блоки на странице, (2) нарисуйте основные дуги, представляющие ограничения, (3) нарисуйте внешние дуги и (4) нарисуйте все оставшиеся дуги. Со временем накопленный опыт позволит вам отойти от этой процедуры и изображать блоки и дуги в соответствии с той идеей, которую вы хотите воплотить в диаграмме.

Правильное расположение блоков является самым важным этапом построения диаграммы. Блоки располагаются в соответствии с их доми-

нированием (по степени важности или по порядку следования). Самый доминантный блок обычно располагается в верхнем левом углу, а наименее доминантный – в нижнем правом. Это приводит к расположению, при котором более доминантные блоки ограничивают менее доминантные, образуя «ступенчатую» схему. Доминирование имеет важнейшее значение для ясного представления процесса. Например, не имеет смысла говорить о контроле над выполнением задания до изготовления детали.

Затем изображают основные дуги, представляющие ограничения. Это является второй важной частью построения диаграммы А0. Они дают основание для разбиения объекта диаграммы, в нашем примере, на 3 системные функции, изображаемые блоками. Рисуя эти дуги, проверяйте, действительно ли каждая из них оказывает влияние, соответствующее декомпозиции объекта. Проследите по списку данных, не отсутствуют ли какие-то дуги, представляющие ограничения. Если это так, вы, возможно, захотите проверить правильность декомпозиции.

Основными дугами, представляющими ограничения, всегда являются внешние дуги, т.е. дуги, представляющие данные, поступающие из непосредственного окружения диаграммы.

Следующим шагом в построении диаграммы является размещение остальных внешних дуг. Таким образом, все данные, входящие в систему или выходящие из нее, оказываются учтенными на рисунке. Потеря внешней дуги – это ошибка интерфейса, одна из самых распространенных в системном анализе. Занимаясь декомпозицией объекта, можно забыть об интерфейсных данных, потому что очень легко сосредоточиться на деталях. Начиная с изображения всех внешних дуг, вы повысите точность диаграммы, включив все интерфейсные данные. И, наконец, нарисуйте все остальные дуги, отражающие детали работы системы в целом. Во-первых, нарисуйте оставшиеся ограничения, действующие между блоками. Во-вторых, нарисуйте основной поток данных. В-третьих, рассмотрите все «патологические» потоки данных (случаи возникновения ошибок). В-четвертых, уточните обратные связи в

потоках данных. В заключение изобразите все обратные связи, вызываемые ошибочными ситуациями.

Здесь следует обратиться к одному очень важному моменту моделирования. На практике оказывается невозможным нарисовать диаграмму сразу набело. Для того чтобы придать некоторую форму данным и функциям, лучше всего сделать набросок (черновик). В процессе работы с черновиком, ситуация начинает проясняться. То, что вначале виделось смутно, становится четким по окончании наброска. При этом часто приходится переименовывать дуги и блоки, зачеркивать дуги, перемещать блоки. Поэтому рекомендуется вначале делать набросок диаграммы, а потом перерисовывать диаграмму набело, чтобы уточнить свое понимание, прояснить ситуацию и создать описание, которое могут посмотреть другие.

Обобщение является последним важным шагом начального этапа моделирования. Вспомните, что для любой SADT-диаграммы есть родительская диаграмма, содержащая ее контекст, где под контекстом понимается блок с набором входных дуг, дуг управления и выходных дуг. Верхняя диаграмма модели (т.е. диаграмма А0) не составляет исключения. Контекстом для нее служит диаграмма А-0, представляющая собой обобщение всей модели. Диаграмма А-0 имеет несколько предназначений. Во-первых, она объявляет общую функцию всей системы. Например, блок на рис. 2.3 с названием *разработать программный продукт* ясно указывает, что делает группа разработчиков. Во-вторых, она дает множество основных типов или наборов данных, которые использует или производит система. Например, *системные требования* позволяют осуществлять контроль над персональными компьютерами, на базе которых разрабатывается программное обеспечение. В-третьих, А-0-диаграмма указывает взаимоотношения между основными типами данных, проводя их разграничение. Например, *исследование рынка* рассматривается как входное данное, нечто, изменяемое процессом, в то время как *системные требования* контролируют выполнение заданий для

определенного типа ЭВМ. Таким образом, А-0-диаграмма представляет собой общий вид изучаемой системы.

При создании диаграммы А-0 используется информация, уже зафиксированная на диаграмме А0. Вначале в центре SADT-бланка рисуют один большой блок, название которого совпадает с названием диаграммы А0. В этот момент следует проверить, адекватно ли название отражает то, что делает система. Все внешние дуги диаграммы А0 изображаются на диаграмме А-0 входящими в соответствующую сторону блока. При этом проверьте, что название каждой дуги описывает то, чем обмениваются система и ее среда. После того как вы изобразите входные и выходные дуги, остановитесь на минуту, чтобы проверить точность описания потока данных. Нарисовав дуги управления, убедитесь, что именно они управляют тем, как система преобразует входные данные в выходные. Наконец, напишите цель и точку зрения модели под основным блоком и сверьте их с тем, что представляется блоком и его дугами.

В процессе обобщения вы убедитесь в том, что он помогает прояснить описание системы, потому что при обобщении вы просматриваете метки дуг для более точного наименования данных, которыми обмениваются система и ее среда. Кроме того, во время обобщения дуги часто объединяются для упрощения изображения модели. В этом случае дуги разветвляются на слои составляющие на диаграмме А0.

Построение диаграммы А-0 свидетельствует об окончании начального этапа моделирования. К этому моменту сделана первая попытка обобщить и описать основную деятельность системы и показать связь системы с ее средой. Несмотря на ограниченное число описанных деталей, диаграммы А-0 и А0 представляют законченную картину, потому что они отражают все основные входы, управления, выходы и функции системы. Общий вид системы, полученный с помощью диаграмм А-0 и А0, – основная цель аналитика на начальном этапе построения SADT-модели.

2.4. Продолжение моделирования

Начальный этап моделирования включает определение объекта, цели и точки зрения модели, ограничения, накладываемые на объект, построение диаграммы верхнего уровня и ее обобщение, составление списков данных и функций, объединение функций в блоки, формирование с использованием списка данных взаимоотношений между блоками. Продолжение моделирования основывается на тех же методах и выводит модель на следующий уровень детализации. Для этого требуется создать отдельную диаграмму для, возможно, каждого блока диаграммы верхнего уровня, затем построить диаграммы для всех блоков новых диаграмм, и так до тех пор, пока модель не будет описывать объект с нужной для достижения цели степенью детализации. Таким образом, продолжение моделирования является рекурсивным процессом.

Декомпозиция модели похожа на начальный этап моделирования, но проще его. Почему? Потому что при декомпозиции модели аналитик всегда находится в контексте, определенном блоком со своими дугами одной из диаграмм. Эта граница, называемая границей объекта, определена двумя способами. Во-первых, объект, цель и точка зрения каждой новой диаграммы уже определены на диаграмме A0. Во-вторых, каждый блок, декомпозируемый в новую диаграмму, уже является ограниченным объектом. Другими словами, он идентифицирует конкретную функцию и все данные, которые для нее требуются или ею порождаются. Строить диаграмму, исходя из этой информации, проще, потому что список данных создается на основе дуг, входящих в блок и выходящих из него, а также потому, что список функций подробно раскрывает название блока. Процесс декомпозиции ограниченного объекта состоит из следующих шагов:

1. выбор блока диаграммы;
2. рассмотрение объекта, определенного этим блоком;
3. создание новой диаграммы;
4. выявление недостатков новой диаграммы;

5. создание альтернативных декомпозиций;
6. корректировка новой диаграммы;
7. корректировка всех связанных с ней диаграмм.

Шаги 1-3 представляют созидательную часть процесса. Выполняя их, аналитик концентрирует свои усилия, связанные с выявлением новой информации об объекте, на более высоком уровне детализации, чтобы достичь ясности изложения. Шаги 4-7 составляют этап саморецензирования, в ходе которого аналитик, создав новую диаграмму, проверяет, какую она несет информацию и в каком она находится отношении с родительской диаграммой. Затем в созданную диаграмму и соответственно в связанные с ней диаграммы вносятся изменения, чтобы достичь ясности для *других.*

Выбор блока. Декомпозиция начинается с чтения диаграммы А0 и определения самого содержательного блока. Это такой блок, декомпозиция которого выявит многие аспекты диаграммы А0 и будет оказывать большое влияние на будущие декомпозиции других блоков этой диаграммы. При выборе самого содержательного блока следует учитывать доминирование, функциональную сложность и понятность. Лучшим не обязательно будет блок, наиболее трудный для понимания. Лучшим блоком для первой декомпозиции будет тот, который позволит наиболее глубоко проникнуть в суть рассматриваемой системы.

Рассмотрим диаграмму А0 для модели разработки программного продукта (рис. 2.4). На ней три блока: *планирование и проектирование разработки продукта, создание программ и документации, маркетинг.* Остановим свой выбор на первом блоке *планирование и проектирование разработки продукта*. Поняв, как происходит *планирование и проектирование разработки программного обеспечения,* мы, видимо, окажемся в лучшем положении для дальнейшей декомпозиции остальной части модели.

Объект, определяемый блоком. Блок 1, *планирование и проектирование разработки продукта*, становится теперь самостоятельным объектом декомпозиции. Для выполнения этой декомпозиции вначале бегло осмотрим

обобщающую диаграмму (посмотрите, пожалуйста, диаграмму А-0 на рис. 2.5) и вспомним цель и точку зрения модели.

Сделав это, мы увидим, что должны описать блок *планирование и проектирование разработки продукта,* чтобы можно было описать процесс разработки программного продукта с точки зрения руководителя разработки. Далее изучаем блок 2 диаграммы А0 и соединенные с ним дуги, чтобы выявить его особенности.

Затем мы составляем список данных со всех дуг, касающихся блока для того, чтобы не потерять какие-либо интерфейсные данные. Данные могут быть декомпозированы в нутрии блока. Например, механизм – *группа разработки*, декомпозируется на *специалиста по маркетингу*, *постановщика задачи*, *руководителя проекта* и *редактора*. Далее, составляем на основе списка данных список функций, придерживаясь функции, соответствующей блоку верхней диаграммы. Обратите внимание на то, что *планирование маркетинга*, *проектирование интерфейса пользователя*, *проектирование системы*, *организация работы программистов* и *проектирование документации*, по-видимому, действительно являются функциями, выполняемыми при *планировании и проектировании разработки программного обеспечения*.

Стремитесь ограничиваться разумным уровнем сложности при объединении функций и данных: четыре-пять функциональных блоков, как правило, лучше всего. Слишком много данных и функций часто содержат слишком много информации. Это приводит к запутанным диаграммам. Наоборот, небольшое число блоков дает слишком мало, и диаграмма становится почти бесполезной. Если вы уверены, что достигли баланса, проверьте, во всех ли отношениях написанные вами слова адекватны объекту, определенному блоком и его граничными дугами на родительской диаграмме. Теперь у вас есть все необходимое для построения диаграммы.

Создание новой диаграммы. Новая диаграмма строится аналогично диаграммам А0 и А-0. Блоки размещаются в соответствии с доминированием (т.е. согласно взаимным ограничениям блоков), затем создаются основные дуги,

представляющие ограничения, потом внешние и, наконец, внутренние дуги. На рис. 2.5-2.6 показаны следующие шаги работы по декомпозиции функций *планирование и проектирование разработки продукта*, *создание программ и документации*

.

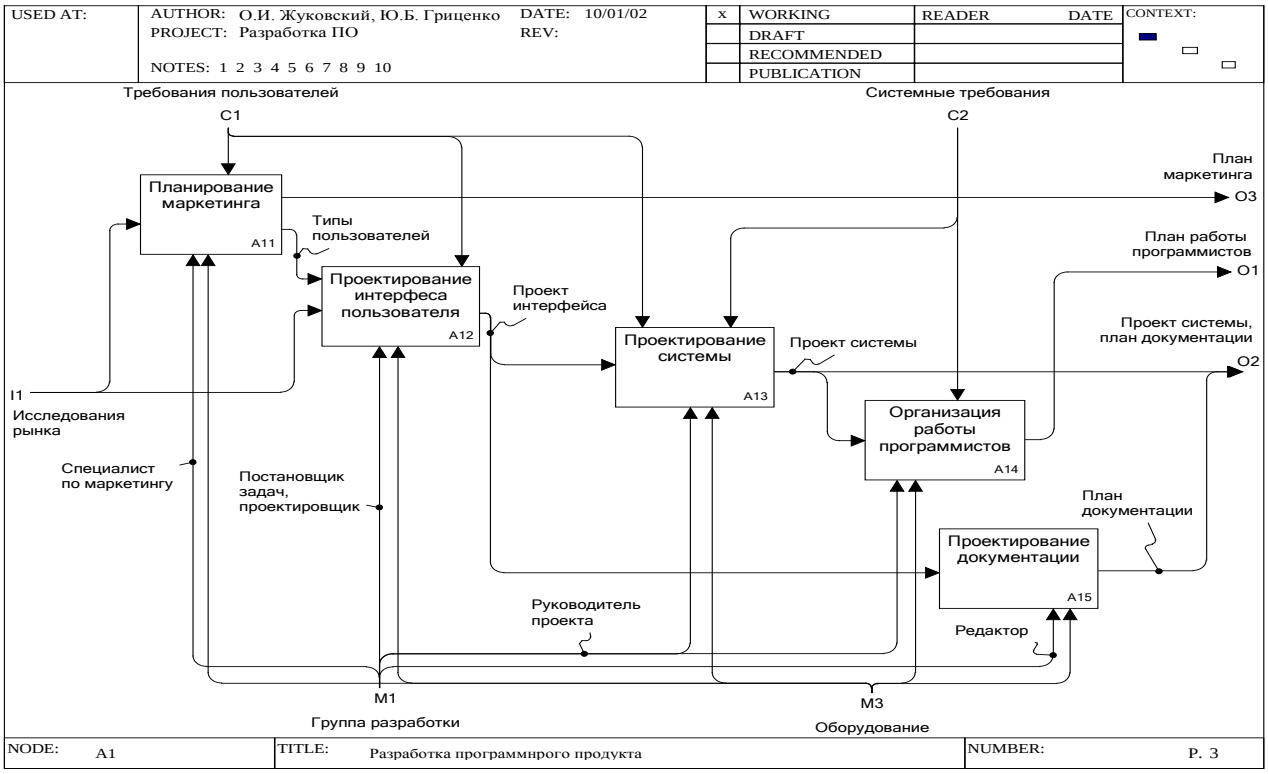


Рис. 2.5. Результат декомпозиции функции – Планирование и проектирование разработки продукта

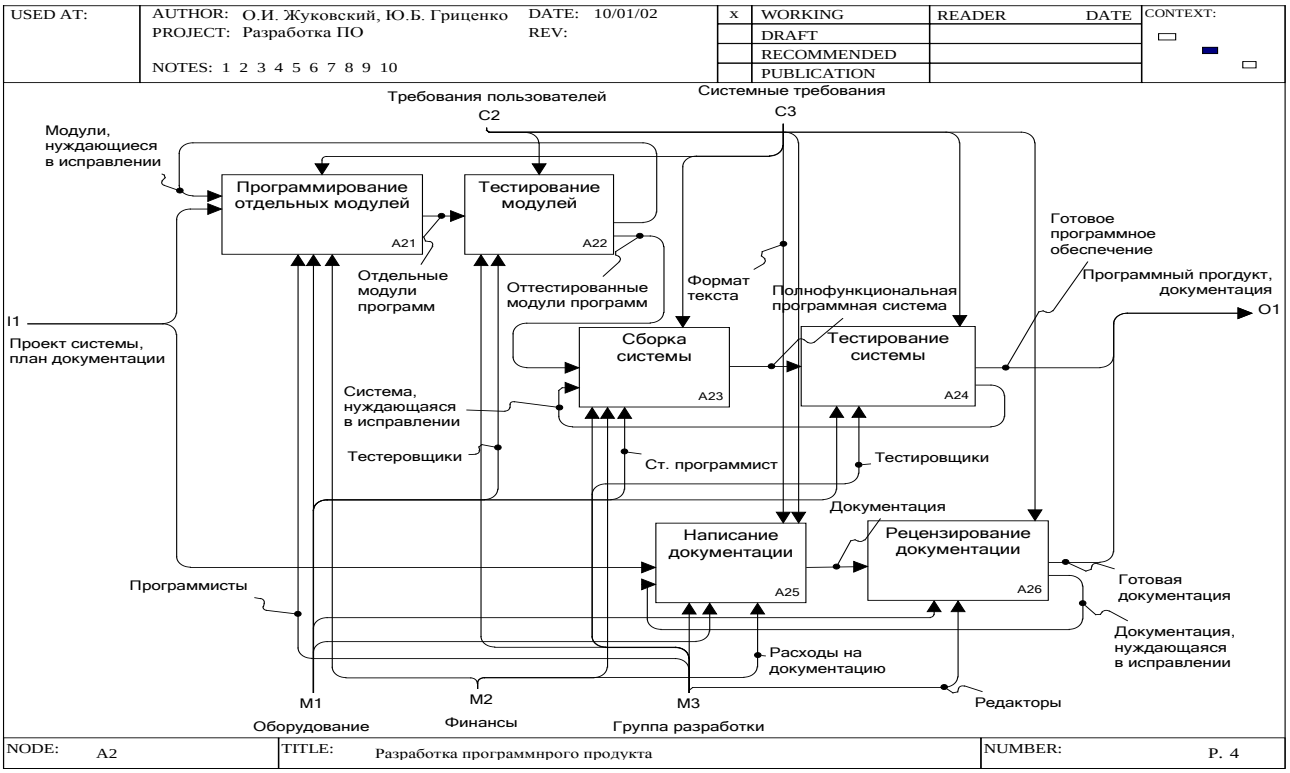


Рис. 2.6. Результат декомпозиции функции – Планирование и проектирование разработки продукта

Инженерам известно, что существенная доля ошибок приходится на интерфейсы. Например, повреждение соединительной прокладки может послужить причиной течи в канализационных трубах, компьютерный кристалл с погнутым выходом может отказать при установке его на монтажную плату, а слишком тугое закрепление клеммы может привести к обрыву электрического провода. Аналогично в моделях сбои часто происходят в точках интерфейса. Для SADT интерфейсными являются места соединения диаграмм со своими родителями. Вот почему каждую декомпозицию необходимо аккуратно соединять со своим родителем.

Дуги выражают связи между блоками. Их вычерчивают не для показа последовательности действий. Они отражают отношения между блоками, независящие от потенциального следования. Такой механизм приводит к

реализации различных сценариев, активизируя блоки в различные моменты времени в зависимости от ситуации.

Может случиться, что, начав рисовать дугу, вы засомневаетесь в том, что она нужна на диаграмме. Не изобразив дугу при первом наброске диаграммы, вы рискуете совершить ошибочный пропуск. Поскольку SADT-диаграммы всегда проверяются другими специалистами, и поскольку отсутствующую дугу никто не сможет обнаружить, изъятие сомнительной дуги с диаграммы – прямая дорога к возникновению ошибок и отсечению дополнительных возможностей. Рекомендуется включать сомнительные дуги в диаграмму с вопросом об их необходимости адресованным читателям. Полученные в ответ комментарии, скорее всего, помогут разрешить ваши сомнения. SADT-дуги представляют собой наборы объектов и поэтому они потенциально могут нести много данных. Например, *оборудование* может быть просто бумагой, а может быть компьютерами, столами и т.д.

Опытные SADT-аналитики не изображают каждый объект отдельной дугой. В крайнем случае, лучше давать определение новой дуги в глоссарии и, возможно, уточнять ее содержимое при декомпозиции тех блоков, которых она касается. Таким образом, вы предоставите читателям достаточно информации для понимания вашей диаграммы и не потратите много времени на избыточно подробное описание модели на слишком ранней стадии, что неизбежно привело бы в дальнейшем к большим переделкам.

2.5. Проверка диаграммы автором

Процесс декомпозиции в SADT сводится к представлению каждого блока диаграммы с помощью диаграммы следующего уровня детализации. Во время декомпозиции составляют список данных и список функций, объединяют функции в блоки и используют список данных при формировании взаимосвязей между блоками. В ходе этого процесса аналитик выполняет такие шаги, как выбор блока, изучение его объекта и построение новой диаграммы. Целью

аналитика на этом этапе является в первую очередь ясное восприятие сути декомпозируемой модели.

Построив диаграмму, попытайтесь самостоятельно выявить ее недостатки, прежде чем рассылать ее для подробного рецензирования. Опытные SADT-аналитики при декомпозиции блока разделяют этап создания и этап критического рассмотрения диаграммы. Они специально выбирают время для того, чтобы окинуть ее критическим взглядом, помня о том, что за несколько минут можно самому обнаружить те ошибки, которые часто выявляются с помощью обратной связи с читателями.

Процесс авторской проверки дает новое направление работе – определение ее качества. На этапе декомпозиции возникает диаграмма, которая декомпозирует блок и его дуги.

Часто в ходе критической оценки выполняют альтернативные декомпозиции, чтобы проверить, является ли исходный набросок лучшим для передачи желаемой информации. Кроме того, проверяют взаимосвязи с родительской и другими диаграммами. После этого во все диаграммы вносятся необходимые изменения.

Иногда у аналитика возникают сомнения относительно блоков диаграммы. На хорошей SADT-диаграмме блоки должны обладать некоторыми важными качествами:

- выполнять строго определенные функции;
- иметь одинаковую сложность;
- иметь одинаковый уровень детализации;
- просто соединяться с другими блоками диаграммы;
- воздействовать на управления, входы и выходы с определенным смыслом;
- работать вместе с другими блоками для выполнения функции диаграммы.

Попробуйте объединить функции и данные иначе или составьте новый список функций, если исходный набор блоков не позволяет осуществит

декомпозицию удачно. Вы можете это сделать и для того, чтобы убедиться в правильности исходного разбиения.

Применяя эти приемы, помните, что критерий качества для блоков достаточно противоречив. Например, добиваясь одинаковой сложности блоков, вы можете усложнить соединения между ними, а упрощение связи между двумя блоками может скрыть какой-либо важный на данном уровне детализации факт. Основным для вас всегда должно быть наилучшее описание декомпозируемого объекта. Построение хороших блоков возможно только при достижении равновесия между требованиями к сложности соединения блоков и к достаточности уровня детализации.

Иногда можно обнаружить две дуги, которые начинаются и кончаются в одних и тех же местах диаграммы. То есть обе дуги начинаются и кончаются у одних и тех же блоков. В этом случае посмотрите на эти две дуги внимательно. Может оказаться, что их следует объединить в одну. Если вы можете придумать хорошее наименование, объединяющее названия этих дуг, объедините их. Если наличие двух дуг имеет определенный смысл, не объединяйте их. Объединение скрывает детали, поэтому не делайте это механически. Исчезновение деталей повредит диаграмме.

Вы можете обнаружить также дугу, описывающую два совершенно различных набора данных. В этом случае изучите дугу, чтобы оценить, приведет ли разделение ее на две к прояснению важных для диаграммы деталей. Будьте очень осторожны и старайтесь сохранить равновесие между стремлением к детализации и сохранением наглядности диаграммы.

Хороший способ оценки диаграммы заключается в рассмотрении сценариев ее работы. Вы представляете себе возможную ситуацию и смотрите, как работает диаграмма в заданных условиях. По мере развития сценария делайте пометки на диаграмме. Это даст вам возможность всегда повторить сценарий, а информация может помочь при декомпозиции блоков этой диаграммы. Таким образом, вы проверите точность и понятность изложенного в диаграмме.

Еще один хороший способ проверки правильности диаграммы – разложение одного – двух ее блоков. (При этом сохраняйте свои наброски, чтобы облегчить будущие декомпозиции этой диаграммы.) Детализация некоторой части новой диаграммы поможет определить сбалансированность декомпозиции и выявить неувязки в распределении функций между новыми блоками.

Корректировка новой диаграммы. Обычно, потратив время на вопросы к диаграмме, тестирование, выполнение альтернативных набросков, автор корректирует диаграмму. В ходе корректировки следите за правильным доминированием, выбором названий блоков, информативностью дуг и делайте пояснения. Помните, что теперь вы рисуете диаграмму, чтобы донести информацию в точном и понятном виде до читательской аудитории.

Для того чтобы составить первое впечатление о выполняемых блоками функциях, читатель вашей диаграммы прочтет по порядку их наименования. Обеспечьте читателям верное первое впечатление от диаграммы, организовав блоки так, чтобы они как можно точнее указывали на взаимное влияние. Один из распространенных приемов для этого заключается в расположении наиболее доминантного блока в верхнем левом углу диаграммы, а наименее доминантного – в нижнем правом. Помните, что расположение блоков может облегчить или затруднить проведение дуг. Иногда вам придется жертвовать расположением блоков в порядке убывания доминантности ради простоты проведения дуг с тем, чтобы получить легко читаемую диаграмму.

Для блоков обычно стараются выбрать содержательные названия. Однако в SADT нет необходимости выражать все с помощью названий блоков, потому что о работе блока многое сообщают метки окружающих его дуг.

Рисуя дуги, старайтесь располагать их аккуратно, минимизируя число пересечений и максимизируя пространство между ними. Правильное графическое расположение вносит большой вклад в повышение наглядности и

понятности диаграммы. Помечайте дуги ясно и точно. Хотя определенное количество слов передает информацию лучше,

Вычерчивая дуги в порядке их значимости, вы сможете оценивать их в процессе рисования и избежите стремления механически присоединять все дуги ко всем блокам.

Закончив построение диаграммы, поясните ее важные аспекты с помощью замечаний или дополнительного материала. Однако будьте осторожны с пояснениями: не используйте их как прикрытие плохого построения диаграмм. Проясняйте только те понятия, которые нельзя изобразить в виде блоков и дуг. Например, пометить дугу M1 на рис. 2.4 словом *оборудование* и сделать замечание, что *оборудование* – это *компьютеры, принтеры, столы, бумага,* хуже, чем просто пометить дугу этим набором слов. С другой стороны, описание типичных незаконченных заданий существенно облегчит объяснение того, как они должны быть завершены.

Исправление взаимосвязанных диаграмм. Создание диаграммы, ответы на связанные с ней вопросы и переделка ее обеспечивают более глубокое понимание родительской диаграммы и диаграмм – потомков вновь построенной диаграммы. Зафиксируйте свое понимание во время исправления диаграммы. (Это как раз тот случай, когда перенесение информации снизу-вверх естественным образом вписывается в технику декомпозиции.) Автор вынужден переносить информацию на другие диаграммы в трех ситуациях: при изменении меток внешних дуг, при появлении новых внешних дуг и при перераспределении функций.

Перенесение необходимо, если изменилось название внешней дуги. Перенесение измененных меток внешних дуг немедленно обеспечивает предоставление родительской диаграммой всех данных, необходимых диаграмме-потомку. Перенесение необходимо также, когда на новой диаграмме появляются новые входные или выходные дуги. Эти новые дуги должны, так или иначе, возникнуть на родительской диаграмме. Есть два пути сделать это: нарисовать новые дуги на родительской диаграмме или объединить дуги новой

диаграммы в одну и изменить соответствующим образом метку дуги на родительской диаграмме. Делая это, соблюдайте правила соединения и разветвления дуг.

Перемещение блоков представляет самую сложную ситуацию. Оно происходит, когда функция (обычно на низком уровне модели) должна появиться, но не появляется на диаграмме, которую вы рисуете, а появляется на другой диаграмме модели, или наоборот. Перенести такую функцию, представленную блоком и всеми его дугами, с одной диаграммы на другую – нелегкое дело. Обычно это приводит к большим изменениям в метках дуг, появлению множества новых и исчезновению некоторых старых дуг. Иногда перемещение одного блока ведет к перемещению других блоков на различные диаграммы, вызывая целую волну изменений. Как правило, перемещение блока влечет за собой обилие технически сложной работы и может привести к ошибкам, если изменения не отслеживаются достаточно тщательно.

2.6. Соглашения по построению диаграмм

В процессе критической оценки содержания диаграммы автор должен оценивать усложненность диаграммы, вызванную неудачным расположением ее блоков и дуг. Соглашения по размещению элементов SADT-диаграмм помогают вычерчивать более читабельные диаграммы, так же как конструкции структурного программирования позволяют писать читабельные программы. В SADT существует несколько типов соглашений по размещению элементов – для блоков, для дуг и для комбинаций блоков и дуг. В этой главе рассматривается каждый из этих типов, описываются правила для них и приводятся соответствующие примеры. По мере обсуждения правил, мы настоятельно рекомендуем внимательно изучать примеры, поскольку они графически иллюстрируют эти правила.

Соглашения по размещению блоков.

- Располагайте блоки по диагонали – от левого верхнего угла диаграммы до правого нижнего, и пронумеруйте их в том же порядке.
- Поместите номер каждого блока в его нижнем правом углу. Стандартное расположение номеров позволяет их быстро находить.
- Запишите С-номер диаграммы, декомпозирующей блок, под правым нижним углом блока. При таком расположении его легко найти. Кроме того, номер блока наглядно связывается с детализирующей его диаграммой.

Соглашения по размещению дуг.

1.Чертите дуги только по вертикали и горизонтали. Таким образом, блоки будут визуально выделяться как точки сбора дуг, которыми блоки и являются. Это помогает также проследить за направлением дуг.

2.Блоки всегда имеют дуги управления и выхода, но могут не иметь входных дуг. Дуги управления накладывают ограничения и включают или выключают функции системы. Без них система не может работать.

3.Если данные служат и для управления, и для входа, вычерчивайте только дугу управления. Этим вы уменьшаете сложность общей картины и делаете очевидным управляющий характер данных.

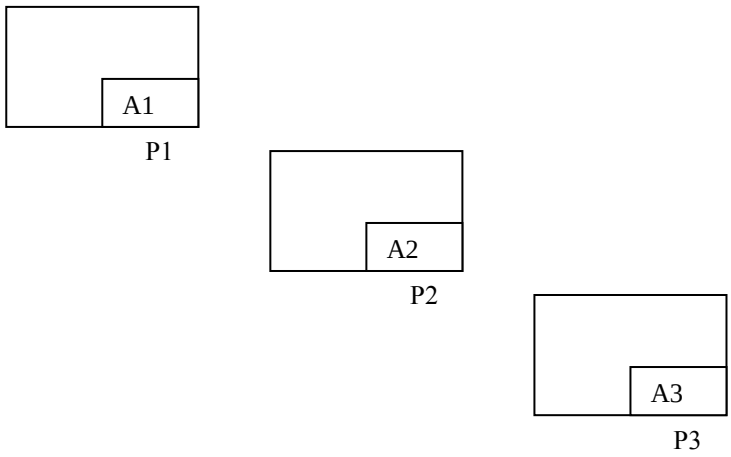


Рис. 2.8. Соглашения по размещению блоков.

4.Максимально увеличьте расстояние между параллельными дугами, оставляя больше места для меток. Это помогает зрительно определять количество дуг и прослеживать их пути (рис. 2.9).

5.Максимально увеличьте расстояние между блоками и поворотами дуг, а также между блоками и пересечениями дуг, чтобы облегчить процесс чтения и уменьшить вероятность перепутать две разные дуги.

6.Объедините дуги, источники которых не указаны на диаграмме, если они представляют одни и те же данные. Этим вы графически покажете единый источник сходных данных (рис. 2.10).

7.Рисуйте циклические обратные связи для одного и того же блока только, чтобы выделить их. Обычно обратную связь изображают на диаграмме, декомпозирующей блок. Однако иногда требуется выделить буферы и повторно используемые объекты (рис. 2.11).

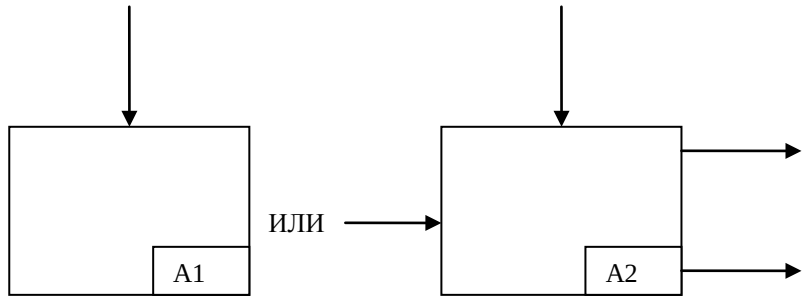


Рис. 2.9. Соглашения по размещению дуг (1-4).

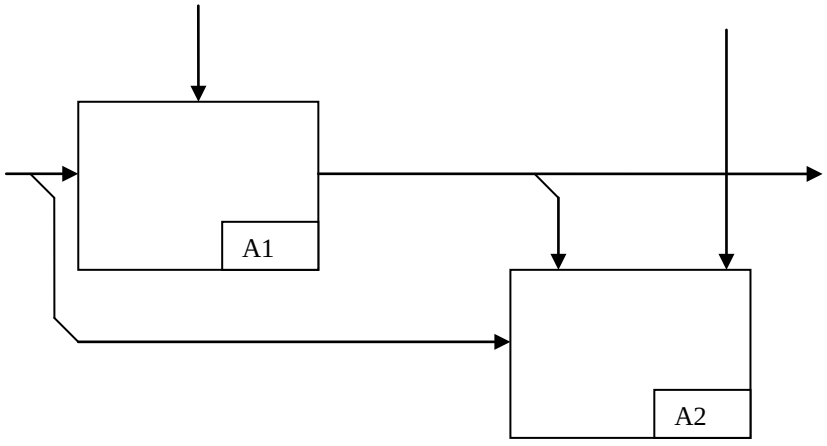


Рис. 2.10. Соглашения по размещению дуг (5-6).

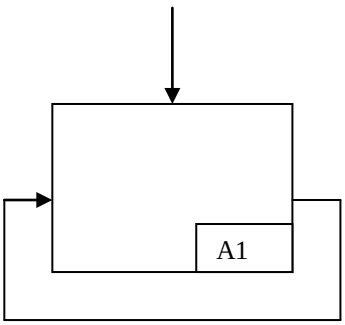


Рис. 2.11. Соглашения по размещению дуг (7).

Соглашения по размещению блоков и дуг.

1. Объединяйте дуги с общим источником и с общим приемником, если они представляют связанные данные. Общее название лучше описывает суть данных (рис. 2.12).

2. Минимизируйте число дуг, касающихся каждой стороны блока, если, конечно, природа данных не слишком разнородна (рис. 2.13).

3. Обратные связи по управлению рисуйте «вверх и над». Таким образом вы покажете ограничивающие обратные связи при минимальном числе линий и пересечений, а также соберете все дуги управления в верхней правой части диаграммы (рис. 2.14).

4. Обратные связи по входу рисуйте «вниз и под». Это позволит показать обратные потоки данных при минимальном числе линий и пересечений, а также собрать все входные дуги в нижней левой части диаграммы (рис. 2.15).

5. Если возможно, присоединяйте дуги к блокам в одной и той же ICOM-позиции. Соединения дуг конкретного типа с блоками будут согласованными, и тем самым вы упростите чтение диаграммы (рис. 2.16).

6. При соединении большого числа блоков избегайте необязательных пересечений дуг. Возможно, это простейшее и самое очевидное правило позволит более всего уменьшить сложность диаграммы (рис. 2.17).

7. Минимизируйте число петель и поворотов каждой дуги. Это также упростит диаграмму (рис. 2.18).

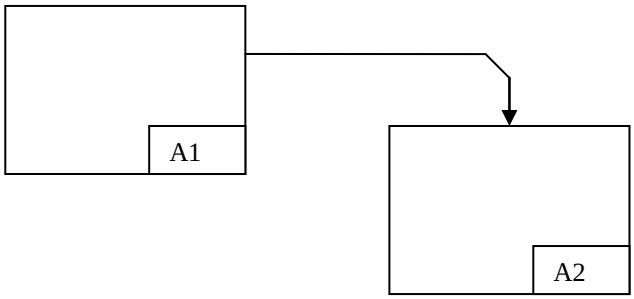


Рис. 2.12. Соглашение по размещению блоков и дуг (1).

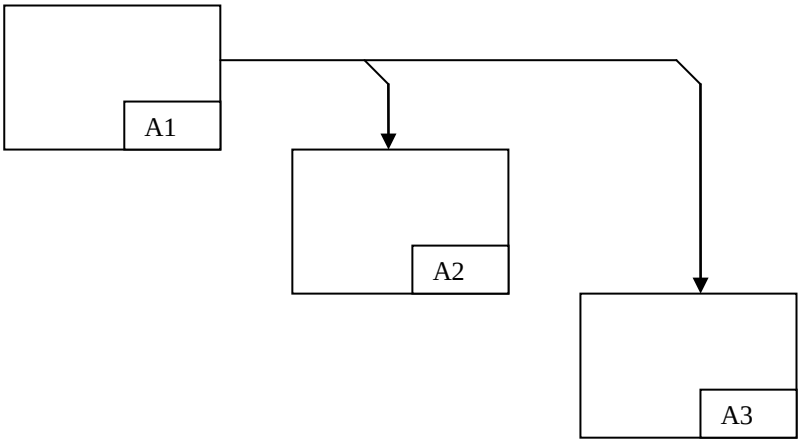


Рис. 2.13. Соглашение по размещению блоков и дуг (2).

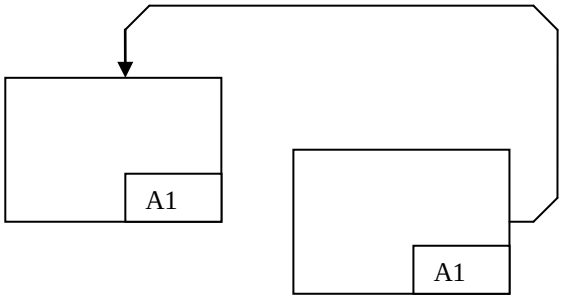


Рис. 2.14. Соглашение по размещению блоков и дуг (3).

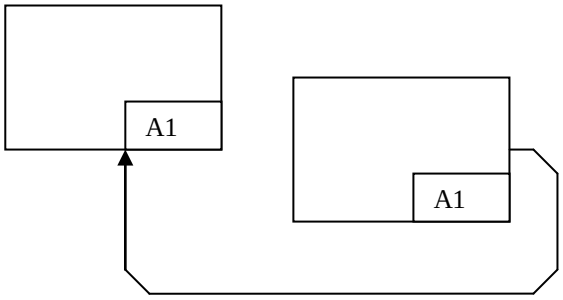


Рис. 2.15. Соглашение по размещению блоков и дуг (4).

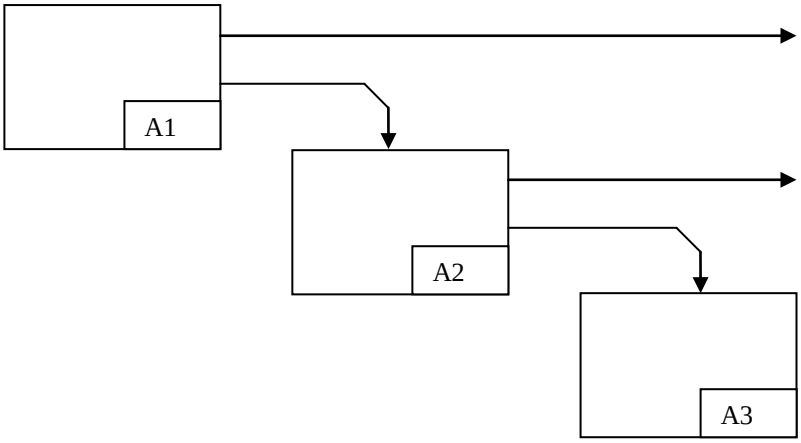


Рис. 2.16. Соглашение по размещению блоков и дуг (5).

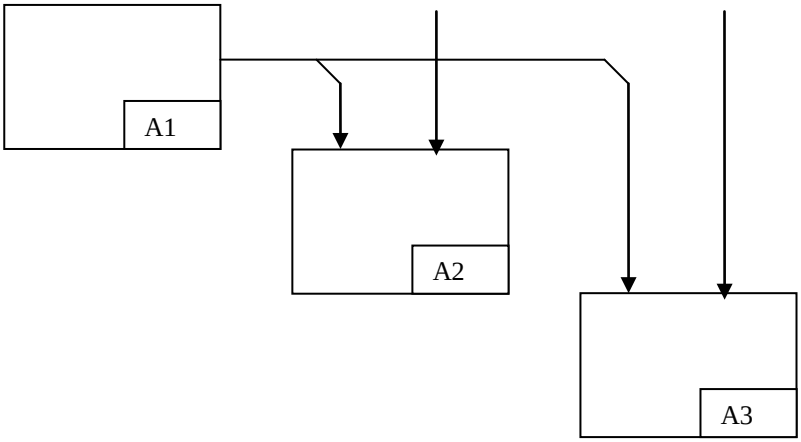


Рис. 2.17. Соглашение по размещению блоков и дуг (6).

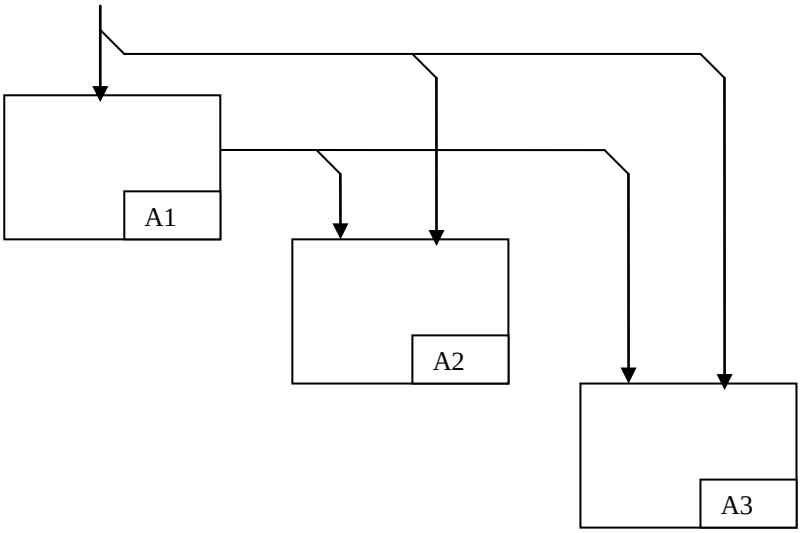


Рис. 2.18. Соглашение по размещению блоков и дуг (7).

2.7. Завершение моделирования

Одна из наиболее частых проблем, возникающих в процессе реализации SADT-проектов, – когда же следует завершить построение конкретной модели? На этот вопрос не всегда легко ответить, хотя существуют некоторые эвристики для определения разумной степени полноты. В этом разделе представлены правила, которыми пользуются опытные SADT-авторы для определения момента завершения моделирования. Однако хотелось бы предупредить, что приведенные здесь правила носят характер рекомендаций. Иногда даже опытные SADT-авторы, применив эти правила, обнаруживают в следствии, что приняли неверное решение. Только длительная практика позволит вам приобрести знания, необходимые для принятия правильного решения об окончании моделирования.

Размер SADT-моделей.

Обычно модель строится слоями, большинство из которых не являются глубокими. Чаще всего ограничиваются тремя уровнями. Опыт показывает, что, как правило, создаются несколько диаграмм второго и третьего уровней только

для того, чтобы убедиться, что для достижения цели уже первый уровень содержит достаточно информации.

Однако типичной также является декомпозиция части SADT-модели на глубину 5-6 уровней. В этом случае на такую глубину декомпозируется обычно один из блоков диаграммы A0. Функции, которые требуют такого уровня детализации, часто очень важны, и их детальное описание дает ключ к секретам работы всей системы. Но хотя важные функции могут нуждаться в глубокой детализации, таких функций при создании одной модели насчитывается, как правило, немного. Модели, обладающие такими функциями, имеют обычно форму зонтика с широким тонким куполом и длинной ручкой, на которой происходит детализация. Поэтому вторая причина, по которой размер SADT-моделей не растет в геометрической прогрессии, заключается в том, что, хотя нередко модель имеет глубину 5-6 уровней, она почти никогда не декомпозируется вся до такой степени детализации.

Прекращение декомпозиции. Как правило, большинством SADT-авторов рекомендуется прекращать моделирование, когда уровень детализации модели удовлетворяет ее цель. Другими словами, вы должны закончить моделирование, когда почувствуете, что дальнейшее продвижение не будет удовлетворять информационные потребности проекта или вступит с ними в противоречие. Хотя интуитивно это правило понятно, ему трудно следовать, не оценив модель. В первое десятилетие использования SADT для создания моделей в различных прикладных областях были разработаны некоторые критерии для определения момента завершения моделирования. Этот опыт показал, что для отдельной модели, которая создается независимо от какой-либо другой модели, декомпозиция одного из ее блоков должна прекращаться, если:

- блок содержит достаточно деталей;
- необходимо изменить уровень абстракции, чтобы достичь большей детализации, блока;

- необходимо изменить точку зрения, чтобы детализировать блок;
- блок очень похож на другой блок той же модели;
- блок очень похож на блок другой модели;
- блок представляет тривиальную функцию.

Эти правила подчеркивают практические аспекты применения SADT для описания систем реального мира с конкретной целью (например, понять работу телефонной станции, чтобы определить требования к ее программному обеспечению).

Одна из типичных ситуаций, встречающихся в конце моделирования – это блок, который описывает систему с нужным уровнем подробности. Проверить достаточность деталей обычно совсем легко. Просто спросите себя, отвечает ли блок на все или на часть вопросов, составляющих цель модели. Если блок помогает ответить на один или более вопросов, то дальнейшая декомпозиция может не понадобиться.

Принятие решения о завершении моделирования. Вероятность принятия неправильного решения о завершении моделирования может быть уменьшена, если вы оцените каждый блок, который хотите детализировать, в соответствии с приведенными выше правилами. Поскольку большинство SADT-моделей обычно содержат от 10 до 320 диаграмм, лучшее время для начала оценки блоков, когда модель достигает этих размеров. Если какая-то часть модели достигла уровня, не требующего дальнейшей декомпозиции, обращайтесь к своему собственному критерию для определения момента завершения моделирования, прежде чем декомпозировать блок, который еще не был детализирован.

3. Автоматизация построения модели

Модель может быть построена как с применением программ, автоматизирующих процесс построения моделей IDEF0 (Design/IDEF, BPWin, ERWin), так и просто любых программ, позволяющих нарисовать диаграммы по правилам IDEF0, например Microsoft Visio.

После знакомства с методическим руководством рекомендуется просмотреть ресурс www.vernikov.ru, являющийся самым полным русскоязычным сводом стандартов по семейству IDEF-технологий (в случае отсутствия времени на доскональное изучение данного ресурса в полном объеме можно сразу перейти к разделу http://vernikov.ru/biznes-modelirovanie/tehnologii-i-standarty.html).

4. План выполнения лабораторной работы и варианты заданий на моделирование

Постройте функциональную модель бизнес-процессов разработки хранилища данных в одной из предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении. При рассмотрении предметной области необходимо выявить не менее трех источников оперативных данных, которые послужат основой для загрузки проектируемого ХД и определят класс задач, решаемых на основе проектируемого хранилища данных. Например, для управления недвижимым имуществом это будут БТИ (Ростехинвентаризация), земельный комитет и Учреждение юстиции по регистрации прав на недвижимое имущество. Для аптечной сети - центральный склад и несколько аптек и т.п.

Выполнение задания рекомендуется проводить согласно следующим этапам:

Сбор информации. Наиболее подходящая стратегия выполнения данного этапа – провести поиск информации об особенностях выбранного процесса в сети Интернет. Если у Вас есть знакомые, имеющие отношения к подобному процессу то опросите их для получения более полного ощущения специфики данного этапа.

Предложите список лиц и документов, которые на ваш взгляд могли бы служить источниками информации при моделировании реального процесса разработки ХД в данной области.

Начало моделирования. Создайте диаграммы A0 и A-0. Обратите внимание (см. раздел 2.2), что эти две диаграммы должны полностью рассказывать все о моделируемом процессе с минимальной степенью детализации. Диаграмма A-0, часто называемая контекстной диаграммой,

определяет все необходимые связи моделируемого процесса с окружающим миром.

В первую очередь создайте диаграмму А0 и обобщив ее создайте диаграмму А-0.

Особое внимание обратите на то, что каждая модель должна иметь определенную цель и создаваться с конкретной точки зрения. Выбор цели осуществляется с учетом вопросов, на которые должна ответить модель, а выбор точки зрения – в соответствии с выбором позиции, с которой описывается система.

Подготовьте список основных типов данных и функций, необходимый для дальнейшей декомпозиции процесса. Начните заполнять словарь данных. Помните, что несколько различных типов данных могут использоваться одной функцией.

При создании диаграммы А0 строго следуйте рекомендациям раздела 2.2. Правильное расположение блоков является самым важным этапом построения диаграммы.

Построенные вами на данном этапе диаграммы А-0 и А0 должны представлять законченную картину процесса создания ХД в выбранной области деятельности, поскольку они отражают все основные входы, управления, выходы и функции моделируемого процесса.

Продолжение моделирования. Продолжение моделирования основывается на тех же методах, что и начальный этап и выводит модель на следующий уровень детализации. Создайте отдельную диаграмму для, возможно, каждого блока диаграммы верхнего уровня, затем постройте для всех блоков новых диаграмм и так до тех пор, пока модель не будет описывать объект с нужной для достижения вашей цели степенью детализации.

Помните, что в результате моделирования вы должны сформировать достаточный для разработки ХД набор документов и определить механизмы их создания.

Не забывайте пополнять и корректировать словарь данных.

Завершение моделирования. Для определения момента завершения моделирования воспользуйтесь указаниями раздела 2.6.

Подготовка отчета. По завершении моделирования подготовьте отчет в формате doc или pdf , содержащий качественное описание области выбранной вами бизнеса с указанием использованных источником информации и набор диаграмм IDEF0, представляющий функциональную модель процесса разработки Хранилища Данных для выбранной вами области бизнеса и содержащую основные бизнес-процессы разработки хранилища данных, представленные в данных методических указаниях. В случае использования программ, автоматизирующих построение функциональной модели, файлы моделей, полученные данными программами, в отчете не предоставляются.

Варианты контрольной работы

Выбор варианта контрольной работы осуществляется по общим правилам с использованием следующей формулы:

V = (N*K) div 100,

где V — искомый номер варианта,

N — общее количество вариантов,

div — целочисленное деление,

при V= 0 выбирается максимальный вариант,

K — значение 2-х последних цифр пароля.

Варианты предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении:

1. Управление недвижимым имуществом города.

2. Управление недвижимым имуществом региона
3. Производство нефтехимической продукции.
4. Торговля автомобилями в регионе
5. Фармацевтическое производство на уровне региона.
6. Ведение аптечной сети.
7. Ведение региональной сети общественного питания.
8. Книготорговля на уровне республики.
9. Ведение сети продовольственных магазинов.
10. Ведение сети косметических магазинов.

Список литературы

1. Туманов В.Е. Проектирование хранилищ данных для систем бизнес-аналитики: учебное пособие — М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2010.
2. Марка Д.А., Клемент МакГоуэн К. Методология структурного анализа и проектирования. – М.:МетаТехнология, 1993 www.interface.ru/case/sadt.htm
3. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998.
4. Калянов Г.Н. Консалтинг при автоматизации предприятий: Научно-практическое издание. Серия «Информатизация России на пороге XXI века». – М.:СИНТЕГ, 1997.
5. Калянов Г.Н. CASE-технологии. Консалтинг при автоматизации бизнес-процессов. – М.:Горячая линия, 2000.