

Лекция 5

План лекции 5

4 МОДЕЛЬ СИСТЕМЫ УПРАВЛЕНИЯ КАК ИНСТРУМЕНТ АНАЛИЗА ПРЕДМЕТНОЙ ОБЛАСТИ

- 4.1 Общая характеристика подходов к моделированию бизнес-процессов
- 4.2 Стандарт реализации функциональной методики IDEF0
- 4.3 Моделирование информационных потоков. Диаграмма потоков данных
- 4.4 Сравнение функционального и объектного подхода к моделированию предметной области
- 4.5 Модели AS-IS и TO-BE. Пример построения

Литература

4 МОДЕЛЬ СИСТЕМЫ УПРАВЛЕНИЯ КАК ИНСТРУМЕНТ АНАЛИЗА ПРЕДМЕТНОЙ ОБЛАСТИ

4.1 Общая характеристика подходов к моделированию бизнес-процессов

Процесс бизнес-моделирования (моделирования процессов), широко используемый как в прошлом, так и в настоящее время, в практике разработки ИС, может быть реализован в рамках различных методик (см., например, [6, 7, 9]).

Цель всех подходов – получить формализованное описание предметной области, которое позволяет не только выявить автоматизируемые функции и документооборот системы, но и способствует совершенствованию ее состава и структуры.

Для построения соответствующих моделей широко используются программные инструменты в виде CASE-средств разработки информационных систем. При этом различные методики моделирования отличаются друг от друга, прежде всего, «взглядом» на вопрос: что представляет собой моделируемая система организационного управления (или как видится модель системы управления организации или ее отдельных подразделений). В соответствии с различными представлениями об организации, методики моделирования процессов принято делить на объектные и функциональные (структурные, или «процессные»).

Объектные методики рассматривают моделируемую организацию как набор взаимодействующих объектов – производственных единиц. Объект (например, в кадровом делопроизводстве) определяется как осязаемая реальность – подразделение или специалист, имеющие четко определяемое поведение (список закрепленных работ). Целью применения данной методики является выделение объектов, составляющих организацию, и распределение между ними обязанностей за выполняемые работы.

Функциональные методики, наиболее известной из которых является методика IDEF, рассматривают организацию как набор функций, преобразующий поступающий поток информации в выходной поток. Процесс преобразования информации потребляет определенные ресурсы. Основное отличие от объектной методики заключается в четком отделении функций (методов обработки данных) от самих данных. Это могут быть функции управления производством, финансами, материально-техническим снабжением и проч., рассматриваемые как процессы.

С точки зрения бизнес-моделирования каждый из представленных подходов обладает своими преимуществами – каждый в своей предметной области. Объектный подход позволяет построить более устойчивую к изменениям систему, лучше соответствует существующим статическим структурам организации. Функциональное моделирование хорошо показывает себя в тех случаях, когда организационная структура рассматривается с учетом процессов, в ней происходящих, либо сама находится в процессе изменения, либо вообще слабо оформлена.

Построение модели, отображающей реальные объекты или процессы, и обсуждение ее с заказчиками проекта достаточно хорошо понимается специалистами системы управления при получении от них информации об их текущей работе.

Контрольные вопросы

1. Что такое структурный (процессный) подход к моделированию деятельности системы организационного управления?
2. Что такое объектный подход к моделированию деятельности системы организационного управления?
3. Области применения объектного и процессного подхода к моделированию?

4.2 Стандарт реализации функциональной методики IDEF0

Широко используемую методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique). Исторически IDEF0 как стандарт был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (IDEF=Icam DEFinition), и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологиям США (NIST).

Целью методики является построение функциональной схемы исследуемой системы, описывающей все необходимые *процессы с точностью, достаточной для однозначного моделирования деятельности системы*.

В основе стандарта IDEF0 лежат четыре основных понятия: *функциональный блок, интерфейсная дуга, декомпозиция, глоссарий*.

Функциональный блок (Activity Box) представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, «производить услуги»). На диаграмме функциональный блок изображается прямоугольником (рис. 21). Этот блок соответствует так называемой *контекстной диаграмме*, которая в ходе моделирования детализируется с учетом иерархического подхода к построению модели, на множество *диаграмм декомпозиции*.

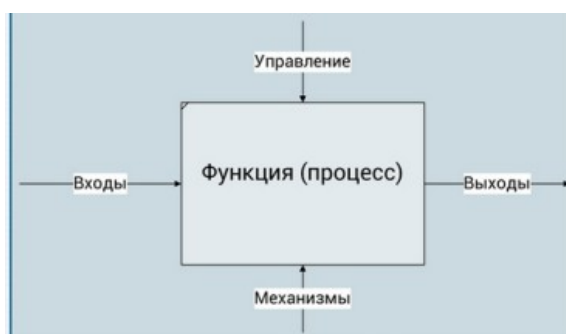


Рисунок 21 – Функциональный блок

Каждая из четырех сторон функционального блока, как на контекстной диаграмме, так и на диаграммах декомпозиции, имеет свое определенное значение (роль), при этом:

- верхняя сторона имеет значение «Управление» (Control), что соответствует передаче на данный уровень нормативно-правовой и плановой информации, являющейся входной для моделируемого процесса;
- левая сторона блока на схеме имеет значение «Вход» (Input), где задаются потоки текущей входной информации;
- правая сторона имеет значение «Выход» (Output), отображающий результатные и промежуточные информационные потоки;
- нижняя сторона имеет значение «Механизм» (Mechanism), что соответствует вектору ресурсов, необходимых для реализации рассматриваемого процесса (трудовые, материальные и другие виды ресурсов).

Как видно из рис. 21, с каждой стороной блока связаны стрелки (в общем случае, их на каждой стороне – несколько). Эти стрелки называют *интерфейсными дугами* (Arrow). Каждая интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют также потоками. С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.). В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей».

Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую. Это и понятно – каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram). На рис. 22 приведен пример функционального блока верхнего уровня управления: контекстная диаграмма исследуемой системы.

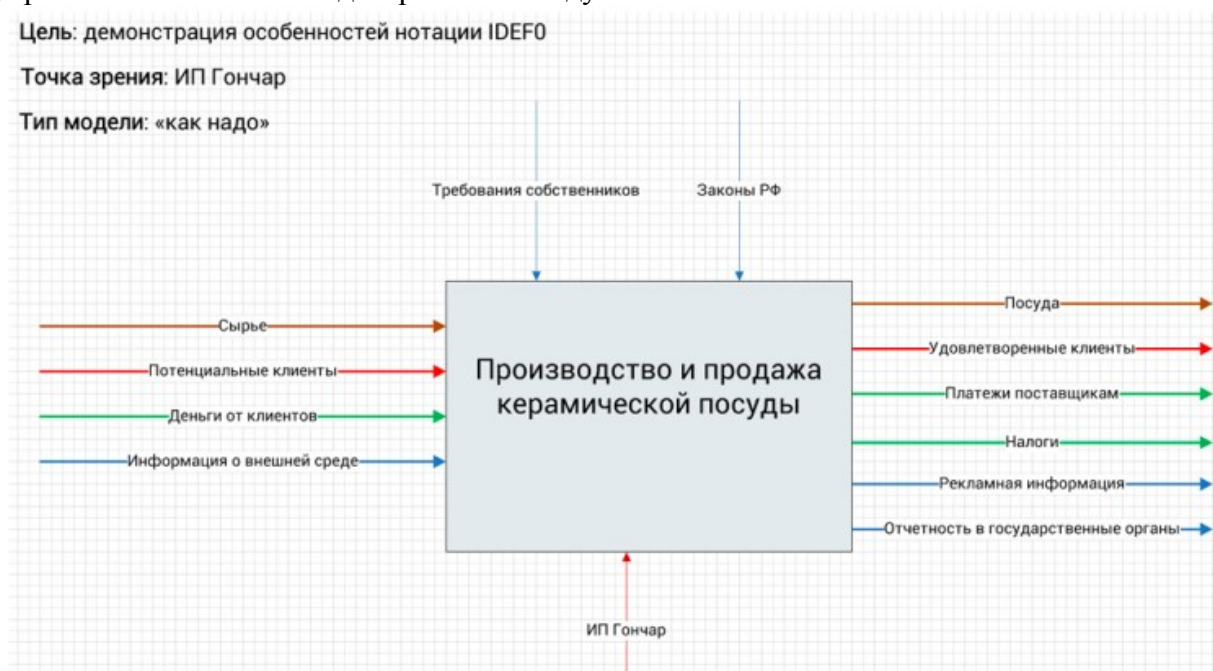


Рисунок 22 – Контекстная диаграмма системы «Производство и продажа керамической посуды»

Декомпозиция (Decomposition) является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели и отражает ту степень подробности в формализованном описании процессов предметной области, которая соответствует целям проектирования (и, следовательно, целям моделирования).

Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Последним из понятий IDEF0 является глоссарий (Glossary). Для каждого из элементов IDEF0 — диаграмм, функциональных блоков, интерфейсных дуг — существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор называется глоссарием и является описанием сущности данного элемента. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Модель IDEF0 всегда начинается с *представления системы как единого целого* — одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком, как отмечалось выше, называется контекстной диаграммой.

В пояснительном тексте к контекстной диаграмме желательно указать цель (Purpose) построения диаграммы в виде краткого описания и зафиксировать точку зрения (Viewpoint) автора данной модели.

Определение и формализация цели разработки IDEF0-модели является крайне важным моментом. Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Ее правильный существенно сокращает временные затраты на построение конечной модели.

Выделение подпроцессов. В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется дочерним блоком — Child Box), или иначе — диаграммой декомпозиции (см. пример на рис. 23).

В свою очередь, функциональный блок — предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит — родительской диаграммой (Parent Diagram). Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. В каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.

Иногда отдельные интерфейсные дуги высшего уровня не имеет смысла продолжать рассматривать на диаграммах нижнего уровня, или наоборот — отдельные дуги нижнего отражать на диаграммах более высоких уровней — это будет только перегружать диаграммы и делать их сложными для восприятия. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования.

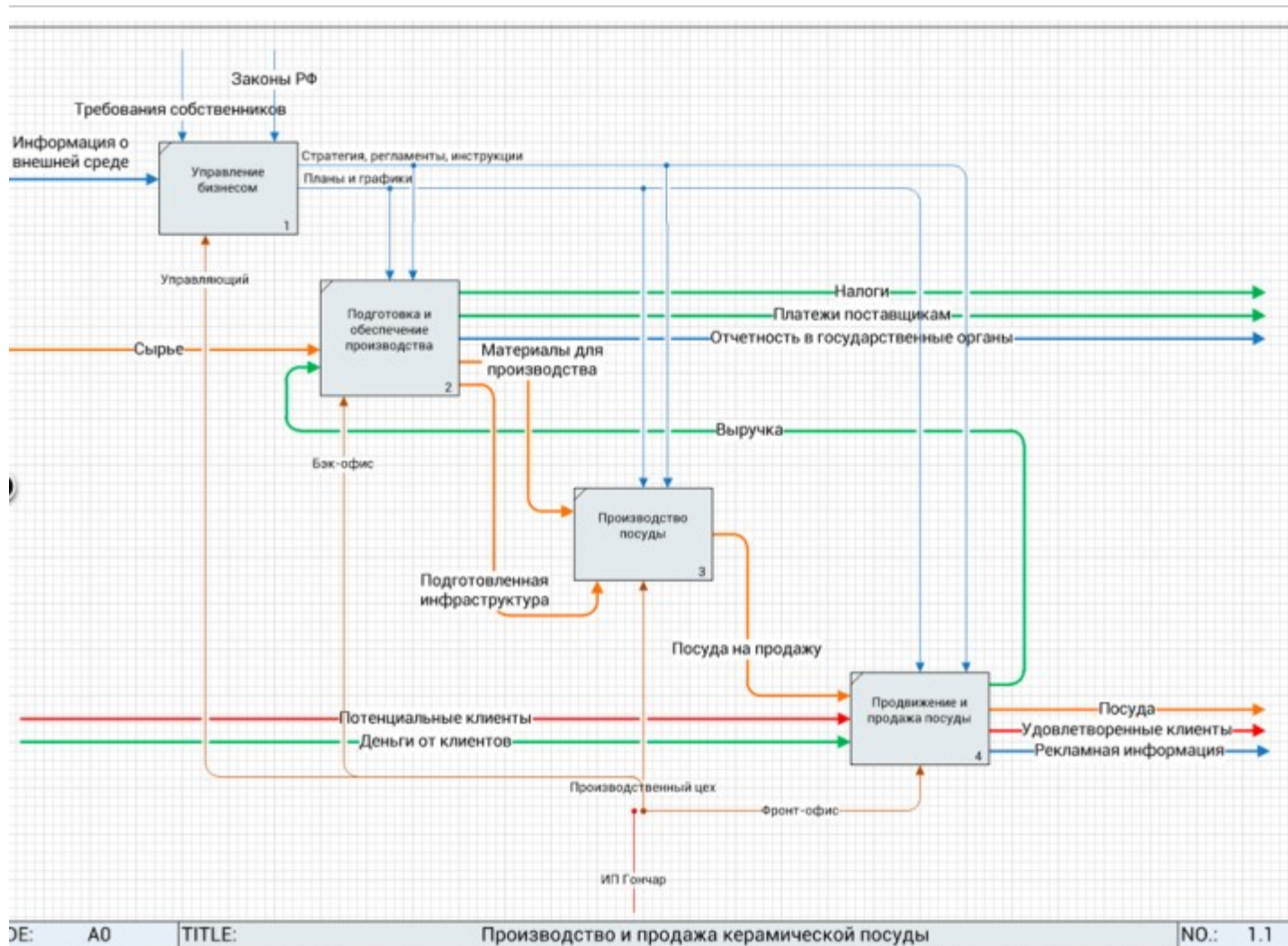


Рисунок 23 – Диаграмма декомпозиции системы «Производство и продажа керамической посуды»

Обозначение «туннеля» (Arrow Tunnel) в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из «туннеля») только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца (стрелки) интерфейсной дуги в непосредственной близости от блока–приемника означает тот факт, что в дочерней по отношению к этому блоку диаграмме эта дуга отображаться и рассматриваться не будет. Чаще всего бывает, что отдельные объекты и соответствующие им интерфейсные дуги не рассматриваются на некоторых промежуточных уровнях иерархии, – в таком случае они сначала «погружаются в туннель», а затем при необходимости «возвращаются из туннеля».

Обычно IDEF0-модели несут в себе сложную и концентрированную информацию, и для того, чтобы ограничить их перегруженность и сделать удобочитаемыми, в стандарте приняты соответствующие ограничения сложности. Рекомендуется представлять на диаграмме от трех до шести функциональных блоков, при этом количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг предполагается не более четырех. В противном случае модель станет достаточно сложной для ее понимания, а следовательно, для проверки ее адекватности реальным или предполагаемым («нормативным») процессам.

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

Создание модели группой специалистов, относящихся к различным сферам деятельности предприятия. Эта группа в терминах IDEF0 называется авторами (Authors). Построение первоначальной модели является динамическим процессом, в течение которого авторы *обязательно опрашивают компетентных лиц* о структуре различных процессов, создавая модели деятельности подразделений. При этом их интересуют ответы на следующие вопросы:

Что поступает в подразделение «на входе»?

Какие функции и в какой последовательности выполняются в рамках подразделения?

Кто является ответственным за выполнение каждой из функций?

Чем руководствуется исполнитель при выполнении каждой из функций?

Что является результатом работы подразделения (на выходе)?

На основе имеющихся положений, документов и результатов опросов создается *черновик* (Model Draft) модели.

Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким кругом компетентных лиц (в терминах IDEF0 — читателей) на предприятии. При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору. Автор, в свою очередь, также письменно соглашается с критикой или отвергает ее с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

Утверждение модели. Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности. Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Наглядность графического языка IDEF0 делает модель вполне читаемой и для лиц, которые не принимали участия в проекте ее создания, а также эффективной для проведения показов и презентаций. В дальнейшем на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений в модели.

Контрольные вопросы

1. Дайте общую характеристику стандарта, реализованного графическим языком IDEF0. Какова цель использования данного стандарта при проектировании ИС?
2. Дайте определения основных понятий: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий. Приведите примеры
3. Что такое контекстная диаграмма и диаграмма декомпозиции?
4. Что такое Model Draft? Зачем и каким образом создается эта модель?
5. Какова роль специалистов организации (предприятия) при создании моделей с использованием стандарта IDEF0? Обоснуйте Ваш ответ.

4.3 Моделирование информационных потоков. Диаграмма потоков данных

Целью методики является построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram — DFD), обеспечивающей правильное описание выходов (отклика системы в виде данных) при заданном воздействии на вход системы (подаче сигналов через внешние интерфейсы). Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

При создании диаграммы потоков данных используются четыре основных понятия: потоки данных, процессы (работы) преобразования входных потоков данных в выходные, внешние сущности, накопители данных (хранилища), отображенных в примере на рис. 24.

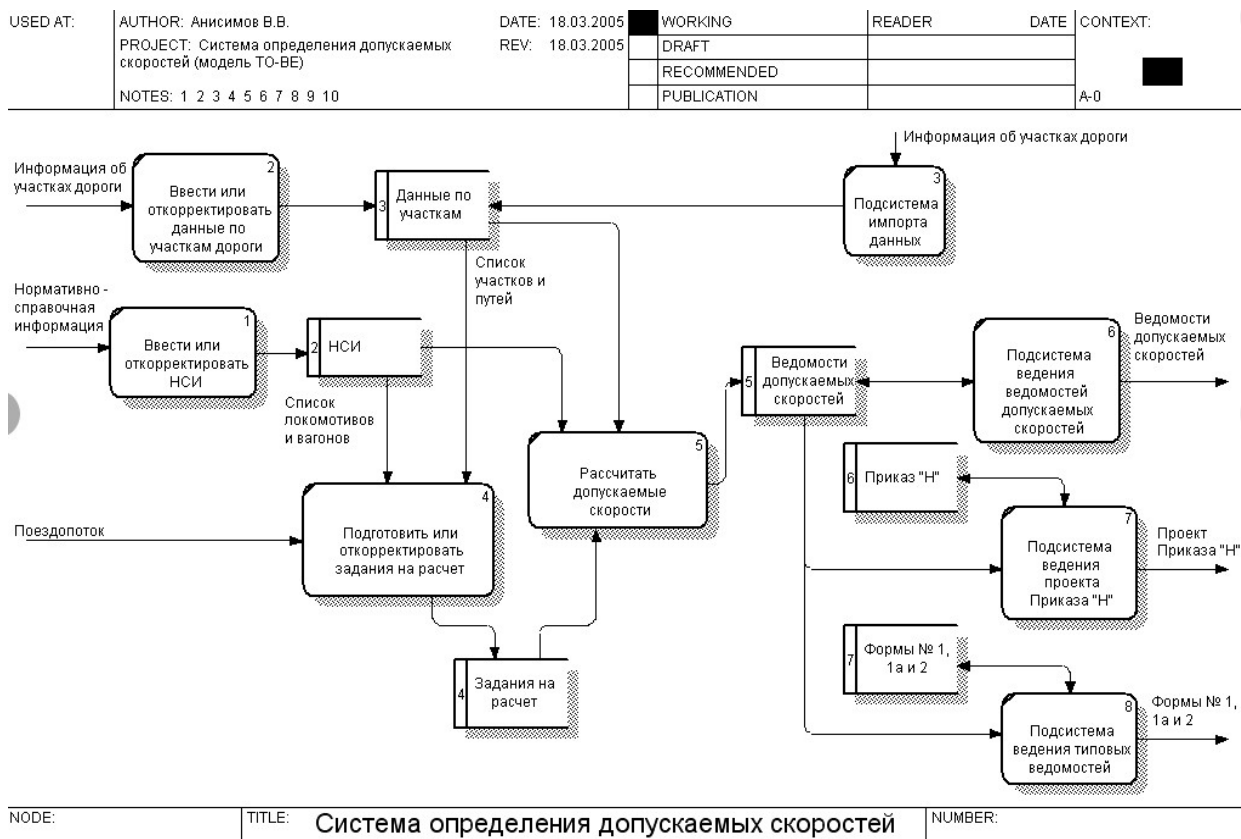


Рисунок 24 – Диаграмма потоков данных системы определения допускаемых скоростей

Потоки данных являются абстракциями, использующимися для моделирования передачи информации (или физических компонент) из одной части системы в другую.

Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Назначение процесса (работы) состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «получить документы по отгрузке продукции»). Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Хранилище (накопитель) данных позволяет на указанных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержимое и быть существительным.

Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником системных данных. Ее имя должно содержать существительное, например, «склад товаров». Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.

Кроме основных элементов, в состав DFD входят словари данных и мини-спецификации.

Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая групповые и индивидуальные потоки данных, хранилища и процессы, а также все их атрибуты.

Мини-спецификации обработки — описывают DFD-процессы нижнего уровня. Фактически мини-спецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы.

Процесс построения DFD начинается с создания так называемой основной диаграммы типа «звезда», на которой представлен моделируемый процесс и все внешние сущности, с которыми он взаимодействует. В случае сложного основного процесса он сразу представляется в виде декомпозиции на ряд взаимодействующих процессов. Критериями сложности в данном случае являются: наличие большого числа внешних сущностей, многофункциональность системы, ее распределенный характер. Внешние сущности выделяются по отношению к основному процессу. Для их определения необходимо выделить поставщиков и потребителей основного процесса, т.е. все объекты, которые взаимодействуют с основным процессом. На этом этапе описание взаимодействия заключается в выборе глагола, дающего представление о том, как внешняя сущность использует основной процесс или используется им. Например, основной процесс — «учет обращений граждан», внешняя сущность — «граждане», описание взаимодействия — «подает заявления и получает ответы». Этот этап является принципиально важным, поскольку именно он определяет границы моделируемой системы.

Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком. Таблица событий включает в себя наименование внешней сущности, событие, его тип (типичный для системы или исключительный, реализующийся при определенных условиях) и реакцию системы.

На следующем шаге происходит декомпозиция основного процесса на набор взаимосвязанных процессов, обменивающихся потоками данных. Сами потоки не конкретизируются, определяется лишь характер взаимодействия. Декомпозиция завершается, когда процесс становится простым, т.е.:

- процесс имеет два-три входных и выходных потока;

- процесс может быть описан в виде преобразования входных данных в выходные;
- процесс может быть описан в виде последовательного алгоритма.

Для простых процессов строится мини-спецификация – формальное описание алгоритма преобразования входных данных в выходные.

Мини-спецификация удовлетворяет следующим требованиям: для каждого процесса строится одна спецификация; спецификация однозначно определяет входные и выходные потоки для данного процесса; спецификация не определяет способ преобразования входных потоков в выходные; спецификация ссылается на имеющиеся элементы, не вводя новые; спецификация по возможности использует стандартные подходы и операции.

После декомпозиции основного процесса для каждого подпроцесса строится аналогичная таблица внутренних событий.

Следующим шагом после определения полной таблицы событий выделяются потоки данных, которыми обмениваются процессы и внешние сущности. Простейший способ их выделения заключается в анализе таблиц событий. События преобразуются в потоки данных от инициатора события к запрашиваемому процессу, а реакции – в обратный поток событий. После построения входных и выходных потоков аналогичным образом строятся внутренние потоки. Для их выделения для каждого из внутренних процессов выделяются поставщики и потребители информации. Если поставщик или потребитель информации представляет процесс сохранения или запроса информации, то вводится хранилище данных, для которого данный процесс является интерфейсом.

После построения потоков данных диаграмма должна быть проверена на полноту и непротиворечивость. Полнота диаграммы обеспечивается, если в системе нет «повисших» процессов, не используемых в процессе преобразования входных потоков в выходные. Непротиворечивость системы обеспечивается выполнением наборов формальных правил о возможных типах процессов: на диаграмме не может быть потока, связывающего две внешние сущности – это взаимодействие удаляется из рассмотрения; ни одна сущность не может непосредственно получать или отдавать информацию в хранилище данных – хранилище данных является пассивным элементом, управляемым с помощью интерфейсного процесса; два хранилища данных не могут непосредственно обмениваться информацией – эти хранилища должны быть объединены.

К преимуществам методики DFD относятся:

- возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы;
- возможность проектирования сверху вниз, что облегчает построение модели «как должно быть»;
- наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы.

К недостаткам модели отнесем: необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потоки) и управляющие процессы с точки зрения DFD ничем не отличаются от обычных; отсутствие понятия времени, т.е. отсутствие анализа временных промежутков при преобразовании данных (все ограничения по времени должны быть введены в спецификациях процессов).

Контрольные вопросы

1. Дайте общую характеристику методики DFD. Приведите основные понятия, используемые при построении диаграммы потоков. Постройте свой пример диаграммы.
2. Что такое словари данных и мини-сценарии? Приведите примеры.
3. Опишите технологию построения DFD – диаграмм.
4. Назовите преимущества и недостатки подхода DFD.

4.4 Сравнение функционального и объектного подхода к моделированию предметной области

Принципиальное отличие между функциональным и объектным подходом заключается в способе декомпозиции системы. Объектно-ориентированный подход использует объектную декомпозицию, при этом статическая структура описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Целью методики является построение бизнес-модели организации, позволяющей перейти от модели сценариев использования к модели, определяющей отдельные объекты, участвующие в реализации бизнес-функций.

Концептуальной основой объектно-ориентированного подхода является объектная модель, которая строится с учетом системы принципов. Основными из них являются:

- абстрагирование;
- инкапсуляция;
- модульность;
- иерархия;
- типизация;
- параллелизм;
- устойчивость.

Основными понятиями объектно-ориентированного подхода являются объект и класс.

Объект — предмет или явление, имеющее четко определенное поведение и обладающие состоянием, поведением и индивидуальностью. Структура и поведение схожих объектов определяют общий для них класс. Класс – это множество объектов, связанных общностью структуры и поведения. Следующую группу важных понятий объектного подхода составляют наследование и полиморфизм. Понятие полиморфизма можно интерпретировать, как способность класса принадлежать более чем одному типу. Наследование означает построение новых классов на основе существующих с возможностью добавления или переопределения данных и методов.

Важным качеством объектного подхода является согласованность моделей деятельности организации и моделей проектируемой информационной системы от стадии формирования требований до стадии реализации. По объектным моделям может быть прослежено отображение реальных сущностей моделируемой предметной области (организации) в объекты и классы информационной системы.

Большинство существующих методов объектно-ориентированного подхода включают язык моделирования и описание процесса моделирования. Процесс – это описание шагов, которые необходимо выполнить при разработке проекта. В качестве языка моделирования объектного подхода используется унифицированный язык моделирования UML, который содержит стандартный набор диаграмм для моделирования.

Диаграмма (Diagram) — это графическое представление множества элементов. Чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями) и представляет собой некоторую проекцию реальной системы.

Объектно-ориентированный подход обладает следующими преимуществами. Объектная декомпозиция дает возможность создавать модели меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование объектного подхода существенно повышает уровень унификации разработки и пригодность для повторного использования, что ведет к созданию среды разработки и переходу к сборочному созданию моделей.

Объектная декомпозиция позволяет избежать создания сложных моделей, так как она предполагает эволюционный путь развития модели на базе относительно небольших подсистем. Объектная модель естественна, поскольку ориентирована на естественное и

понятное восприятие мира человеком.

К недостаткам объектно-ориентированного подхода относятся высокие начальные затраты. Этот подход не дает немедленной отдачи. Эффект от его применения сказывается после разработки двух-трех проектов и накопления повторно используемых компонентов. Диаграммы, отражающие специфику объектного подхода для больших систем, менее наглядны.

В функциональных моделях (DFD-диаграммах потоков данных, SADT-диаграммах) главными структурными компонентами являются функции (операции, действия, работы), которые на диаграммах связываются между собой потоками объектов. Несомненным достоинством функциональных моделей является реализация структурного подхода к проектированию ИС по принципу «сверху - вниз», когда каждый функциональный блок может быть декомпозирован на множество подфункций и т.д., выполняя, таким образом, модульное проектирование ИС. Для функциональных моделей характерны процедурная строгость декомпозиции ИС и наглядность представления.

При функциональном подходе объектные модели данных в виде ER-диаграмм «объект — свойство — связь» разрабатываются отдельно. Для проверки корректности моделирования предметной области между функциональными и объектными моделями устанавливаются взаимно однозначные связи.

Главный недостаток функциональных моделей заключается в том, что процессы и данные существуют отдельно друг от друга — помимо функциональной декомпозиции существует структура данных, находящаяся на втором плане. Кроме того, не ясны условия выполнения процессов обработки информации, которые динамически могут изменяться.

Перечисленные недостатки функциональных моделей снимаются в объектно-ориентированных моделях, в которых главным структурообразующим компонентом выступает класс объектов с набором функций, которые могут обращаться к атрибутам этого класса. Для классов объектов характерна иерархия обобщения, позволяющая осуществлять наследование не только атрибутов (свойств) объектов от вышестоящего класса объектов к нижестоящему классу, но и функций (методов).

В случае наследования функций можно абстрагироваться от конкретной реализации процедур (абстрактные типы данных), которые отличаются для определенных подклассов ситуаций. Это дает возможность обращаться к подобным программным модулям по общим именам (полиморфизм) и осуществлять повторное использование программного кода при модификации программного обеспечения. Таким образом, адаптивность объектно-ориентированных систем к изменению предметной области по сравнению с функциональным подходом значительно выше.

При объектно-ориентированном подходе изменяется и принцип проектирования ИС и отдельных функциональных подсистем. Сначала выделяются классы объектов, а далее в зависимости от возможных состояний объектов (жизненного цикла объектов) определяются методы обработки (функциональные процедуры), что обеспечивает наилучшую реализацию динамического поведения информационной системы.

Для объектно-ориентированного подхода разработаны графические методы моделирования предметной области, обобщенные в языке унифицированного моделирования UML. Однако по наглядности представления модели пользователю-заказчику объектно-ориентированные модели явно уступают функциональным моделям, особенно при проектировании комплексных ИС, о чем свидетельствует опыт разработки систем класса ERP.

При выборе методики моделирования предметной области обычно в качестве критерия выступает также степень ее динамичности. Для более регламентированных задач больше подходят функциональные модели, для более адаптивных бизнес-процессов (управления рабочими потоками, реализации динамических запросов к информационным

хранилищам) — объектно-ориентированные модели. Однако в рамках одной и той же ИС для различных классов прикладных задач могут требоваться различные виды моделей, описывающих одну и ту же проблемную область. В таком случае должны использоваться комбинированные модели предметной области.

Итак, каждая из рассмотренных методик позволяет решить задачу построения формального описания рабочих процедур исследуемой системы. При этом все методики позволяют построить модель «как есть» и «как должно быть». С другой стороны, как было сказано выше, каждая из этих методик обладает существенными недостатками. Их можно обобщить следующим образом: недостатки применения отдельной методики лежат не в области описания реальных процессов, а в неполноте методического подхода.

Функциональные методики в целом лучше дают представление о существующих функциях в организации, о методах их реализации, причем чем выше степень детализации исследуемого процесса, тем лучше они позволяют описать систему. Под лучшим описанием в данном случае понимается наименьшая ошибка при попытке по полученной модели предсказать поведение реальной системы. На уровне отдельных рабочих процедур их описание практически однозначно совпадает с фактической реализацией в потоке работ.

На уровне общего описания системы функциональные методики допускают значительную степень произвола в выборе общих интерфейсов системы, ее механизмов и т.д., то есть в определении границ системы. Хорошо описать систему на этом уровне позволяет объектный подход, основанный на понятии сценария использования. Ключевым является понятие о сценарии использования как о сеансе взаимодействия действующего лица с системой, в результате которого действующее лицо получает нечто, имеющее для него ценность. Использование критерия ценности для пользователя дает возможность отбросить не имеющие значения детали потоков работ и сосредоточиться на тех функциях системы, которые оправдывают ее существование. Однако и в этом случае задача определения границ системы, выделения внешних пользователей является сложной.

Технология потоков данных, исторически возникшая первой, легко решает проблему границ системы, поскольку позволяет за счет анализа информационных потоков выделить внешние сущности и определить основной внутренний процесс. Однако отсутствие выделенных управляющих процессов, потоков и событийной ориентированности не позволяет предложить эту методику в качестве единственной.

Наилучшим способом преодоления недостатков рассмотренных методик является формирование синергетической методики, объединяющей различные этапы отдельных методик. При этом из каждой методики необходимо взять часть методологии, наиболее полно и формально изложенную, и обеспечить возможность обмена результатами на различных этапах применения синергетической методики. В бизнес - моделировании неявным образом идет формирование подобной синергетической методики.

Идея синергетической (синтетической) методики заключается в последовательном применении функционального и объектного подхода с учетом возможности реинжиниринга существующей ситуации.

Рассмотрим применение названной методики на примере разработки административного регламента, описывающего технологию реализации комплекса управленческих операций, выполняемых в рамках должностных обязанностей работников системы управления.

При построении административных регламентов выделяются следующие основные стадии.

Определение границ системы. На этой стадии при помощи анализа потоков данных выделяют внешние сущности и собственно моделируемую систему.

Выделение сценариев использования системы. На этой стадии при помощи критерия полезности строят для каждой внешней сущности набор сценариев использования системы.

Добавление системных сценариев использования. На этой стадии определяют сценарии, необходимые для реализации целей системы, отличных от целей пользователей.

Построение контекстной диаграммы (диаграммы активностей) по сценариям использования. На этой стадии строят набор действий системы, приводящих к реализации сценариев использования;

Функциональная декомпозиция диаграмм активностей как контекстных диаграмм методики IDEF0.

Формальное описание отдельных управленческих работ в виде административного регламента (с применением различных нотаций).

Контрольные вопросы

1. Что является концептуальной основой объектно-ориентированного подхода к моделированию предметной области?

2. Дайте понятие объекта и класса. Приведите примеры использования этих понятий при построении модели некоторой предметной области в системе организационного управления.

3. Охарактеризуйте возможности использования унифицированного языка моделирования UML для формализации работ на различных этапах жизненного цикла ИС.

4. Охарактеризуйте функциональный подход к моделированию и дайте понятие ER-диаграмм.

5. Назовите достоинства и недостатки при использовании объектно-ориентированного и функционального подхода применительно к различным предметным областям.

6. Охарактеризуйте методику синергетического моделирования, проиллюстрируйте ее применение.

4.5 Модели AS-IS и TO-BE. Пример построения

Модель AS-IS. После обследования предметной области, как правило, строится модель существующей системы управления AS-IS

Целью построения функциональных моделей обычно является выявление наиболее слабых и уязвимых мест деятельности организации, анализе преимуществ новых бизнес-процессов и степени изменения существующей структуры деятельности организации .

Иначе говоря: анализ недостатков и "узких мест" начинают с построения модели **AS-1S («Как есть»: дескриптивной модели)**, т.е. модели существующей организации работы реальной системы.

Приведем пример описания предметной области и модель AS-IS, ей соответствующую [8].

Пусть рассматривается одна из задач («Учет данных в маршрутном листе») в системе грузовых перевозок некоторой торгово-производственной компании. У компании есть свой парк автомобилей и штатные водители. Одной из повседневных задач компании является перевозка грузов со склада в магазины по определенным маршрутам. Маршруты создает диспетчер. При составлении маршрута диспетчер перечисляет в порядке следования точки маршрута (магазины или склады) и примерное время продолжительности маршрута.

Основная задача – это учет показаний, таких как величина пробега и расход горючего, с дальнейшей передачей их в бухгалтерию. Для этого диспетчер ежемесячно составляет путевой лист, в котором указывает номер путевого листа, дату, данные об автомобиле, данные о водителе, и маршруты на каждый день с временем выезда и временем приезда. По составленным маршрутам водитель должен развести груз. Путевой лист распечатывается, в нем расписывается диспетчер, который составил путевой лист, подписанный документ

выдается водителю. Водитель во время работы заполняет каждый день в выданном ему путевом листе показания спидометра на начало дня и на конец дня. В конце месяца, когда вся работа выполнена, водитель расписывается на путевом листе и приносит его на согласование и утверждение. Согласованный и утвержденный путевой лист передается в бухгалтерию для учета расхода горюче-смазочных материалов (ГСМ).

На практике иногда возникают проблемы, которые связаны со следующими факторами:

- имеющим место перерасходе ГСМ;
- отсутствием обоснованной системы корректировки нормативов использования ГСМ;
- отсутствием оперативности в сдаче путевых листов водителями;
- отсутствием контроля со стороны диспетчера о фактически пройденных автомобилями маршрутах и соответствующем расходе горючего;
- отсутствием на предприятии координации грузоперевозок в рамках системы “Грузоперевозки – бухгалтерский учет затрат”.

Исходя из изложенного выше, будем пользоваться в данной работе следующими терминами.

Автомобиль – транспортное средство для перевозки грузов.

Водитель – работник, имеющий водительское удостоверение и управляющий автомобилем.

Диспетчер – работник, составляющий маршруты движения автомобилей и путевые листы.

Путевой лист – основной первичный документ учета работы автомобиля и водителя, выдаваемый водителю при выходе автомобиля на маршруты и сдаваемый по окончании планового периода в бухгалтерию для учета ГСМ.

Маршрут – последовательность точек планового движения автомобиля (магазин или склад), по которым водитель должен развести груз.

Точка маршрута – адрес магазина или склада, один из пунктов маршрута.



Рисунок 25 – Контекстная диаграмма процесса формирования путевого листа в модели AS-IS

Анализ названных выше процессов предметной области позволил построить диаграмму модели AS-IS («Как есть») с использованием стандарта IDEF0. Первый уровень иерархического процесса формирования путевого листа представлен контекстной диаграммой, приведенной на рисунке 25. На вход процесса подается плановая информация

по датам месяца и оперативные фактические данные, заносимые в путевой лист водителем. В качестве управляющих механизмов используется нормативно-организационная документация предприятия (приказы директора, должностные инструкции сотрудников системы управления, нормативы использования ГСМ). Ресурсы, необходимые для реализации процесса, представлены нижними стрелками диаграммы: диспетчер, водитель, директор, бухгалтер. Результатом процесса является оформленный документ, который называется путевой лист. Это, как отмечалось выше, основной документ, в котором зафиксирована информация о плановых маршрутах движения автомобилях и о фактическом расходе горючего при перевозке грузов.

На рисунке 26 представлена диаграмма декомпозиции процесса формирования путевого листа в виде четырех подпроцессов. Это планирование маршрутов движения автомобиля, занесение оперативной учетной информации, согласование и утверждение путевого листа и оперативный учет путевых листов в бухгалтерии.

В существующих условиях подпроцесс занесения оперативной учетной информации в путевой лист осуществляется самим водителем. На схеме, изображенной на рисунке 27 и реализованной с использованием нотации DFD, отображен циклический процесс занесения водителем учетной информации в путевой лист.

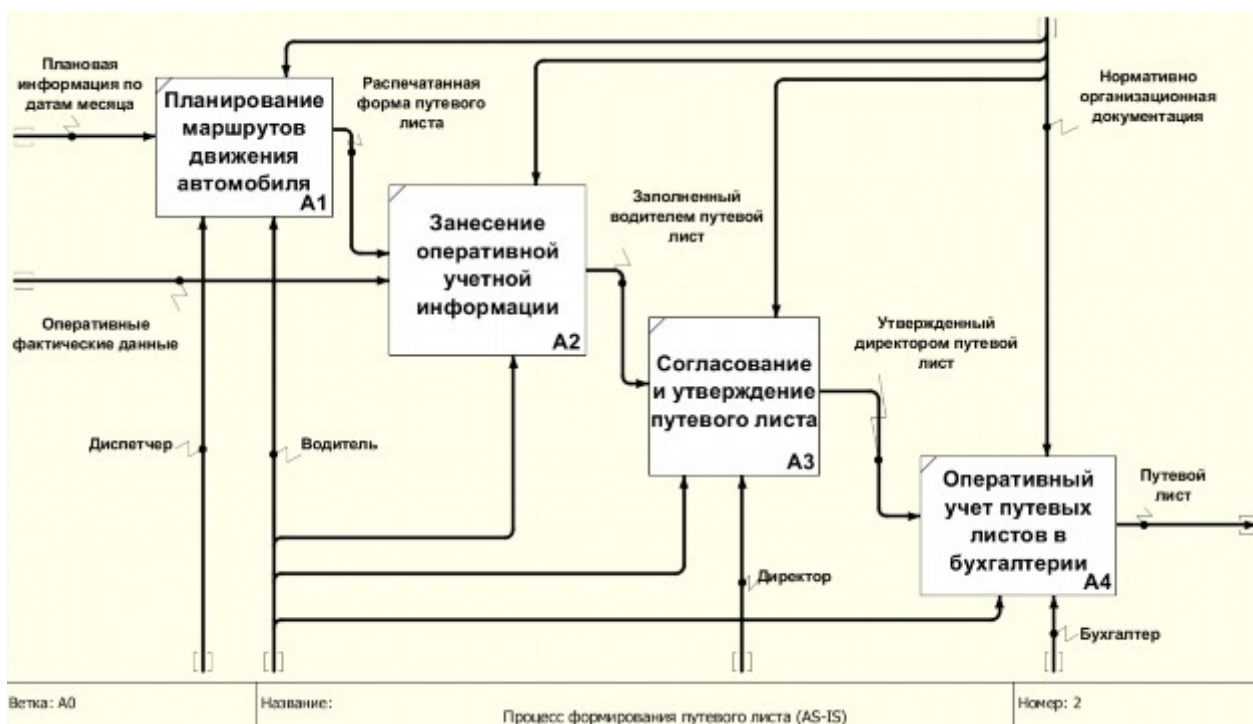


Рисунок 26 – Диаграмма декомпозиции процесса формирования путевого листа в модели AS-IS

По результатам этого процесса водитель приносит диспетчеру путевой лист, который он заполнял в течении месяца. Далее осуществляется процедура корректности заполнения этого документа водителем, после чего он утверждается директором и передается в бухгалтерию.

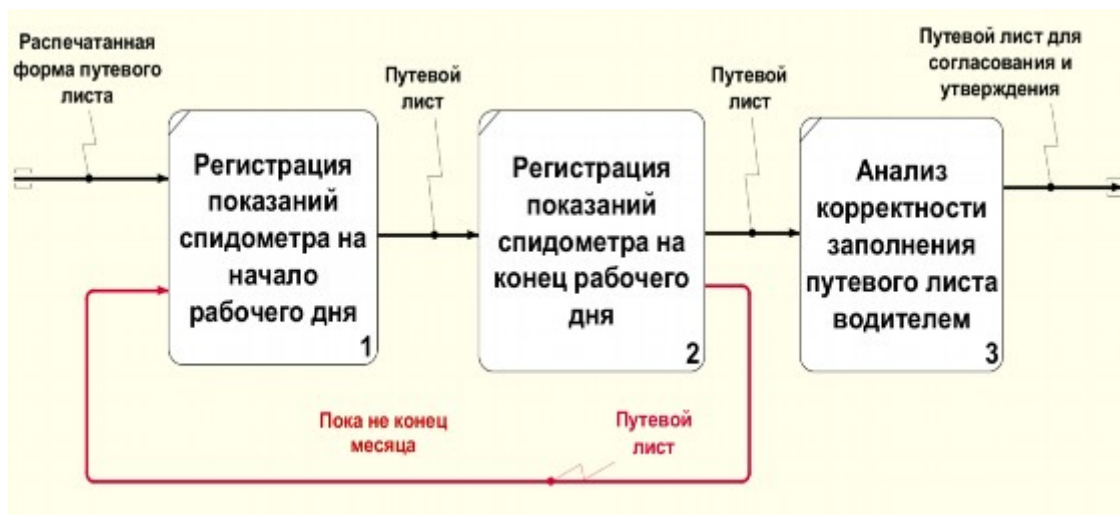


Рисунок 27 – Диаграмма декомпозиции процесса занесения учетной информации путевого листа в модели AS-IS

Таким образом, из модели AS-IS следует, что **путевой лист весь месяц находится у водителя и контроль заполнения этого значимого документа осуществляется лишь в конце месяца перед сдачей его бухгалтеру**. Это приводит к **проблеме** контроля перевозок и расхода топлива. Сотрудники могут использовать служебный транспорт в личных целях, что ведёт к финансовым потерям предприятия. Задача повышения контроля перевозок и расхода топлива является актуальной для каждого предприятия, стремящегося к эффективной экономической деятельности. Решением этой задачи является качественное управление грузовыми перевозками, которое в значительной мере предопределяет рациональное использование топлива. В связи со сказанным предлагается другая организация учета – в рамках модели TO-BE.

Модель TO-BE («Как будет»: нормативная модель). Исправление недостатков в деятельности организации, перенаправление информационных и материальных потоков приводит к созданию модели TO-BE – модели более совершенной организации бизнес-процессов.

Для рассматриваемой предметной области предложен проект с архитектурой системы мониторинга грузовых перевозок, изображенной на рис. 28.

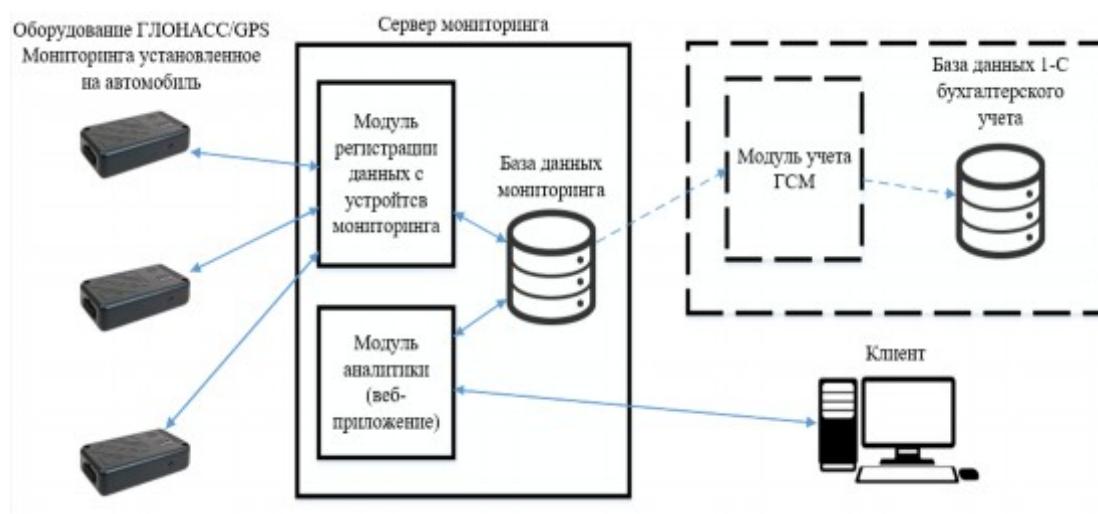


Рисунок 28 – Архитектура системы мониторинга грузовых перевозок

Анализ реализации предложенной архитектуры системы *мониторинга грузовых перевозок* позволил предложить принципиально другой подход к процессу формирования путевого листа.

Предлагается модель рассматриваемого процесса ТО-ВЕ («Как будет»), которая представлена на рис. 29 и рис. 30.

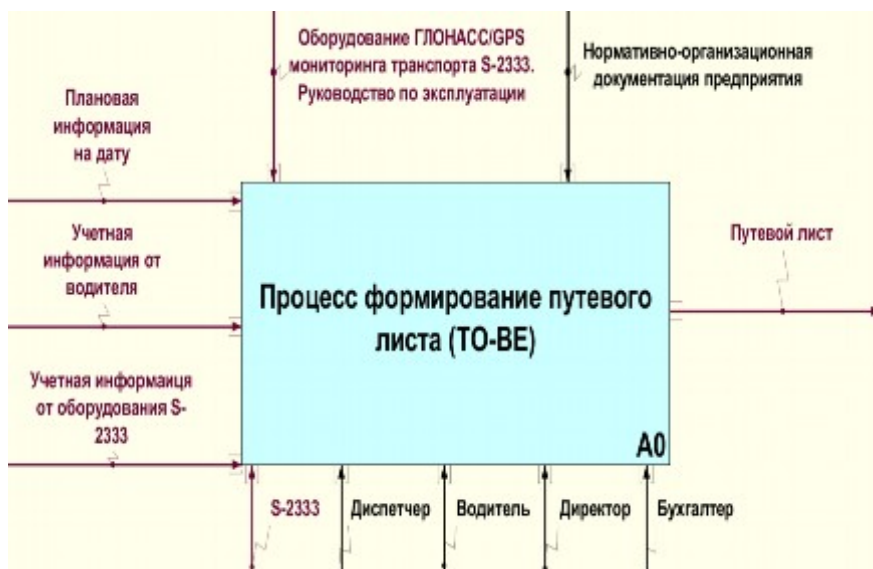


Рисунок 29 – Контекстная диаграмма процесса формирования путевого листа в модели ТО-ВЕ

Как видно из контекстной диаграммы модели ТО-ВЕ, к ресурсам, которые требуются для реализации данного процесса, добавлено **устройство мониторинга s-2333** и другие связанные с этим изменения. На рис. 30 внесенные в модель AS-IS изменения при разработке модели ТО-ВЕ выделены вишневым цветом.

Задачи планирования и учета показателей путевого листа предлагается решать с использованием специальной базы данных, которая в модели названа – база данных мониторинга.

На рис. 30 видны операции процесса формирования путевого листа с использованием базы данных. Это планирование маршрутов движения автомобиля на дату, занесение водителем учетной информации на дату, регистрация оперативной учетной информации с использованием устройства S-2333, согласование путевого листа с диспетчером, утверждение путевого листа и учет ГСМ.

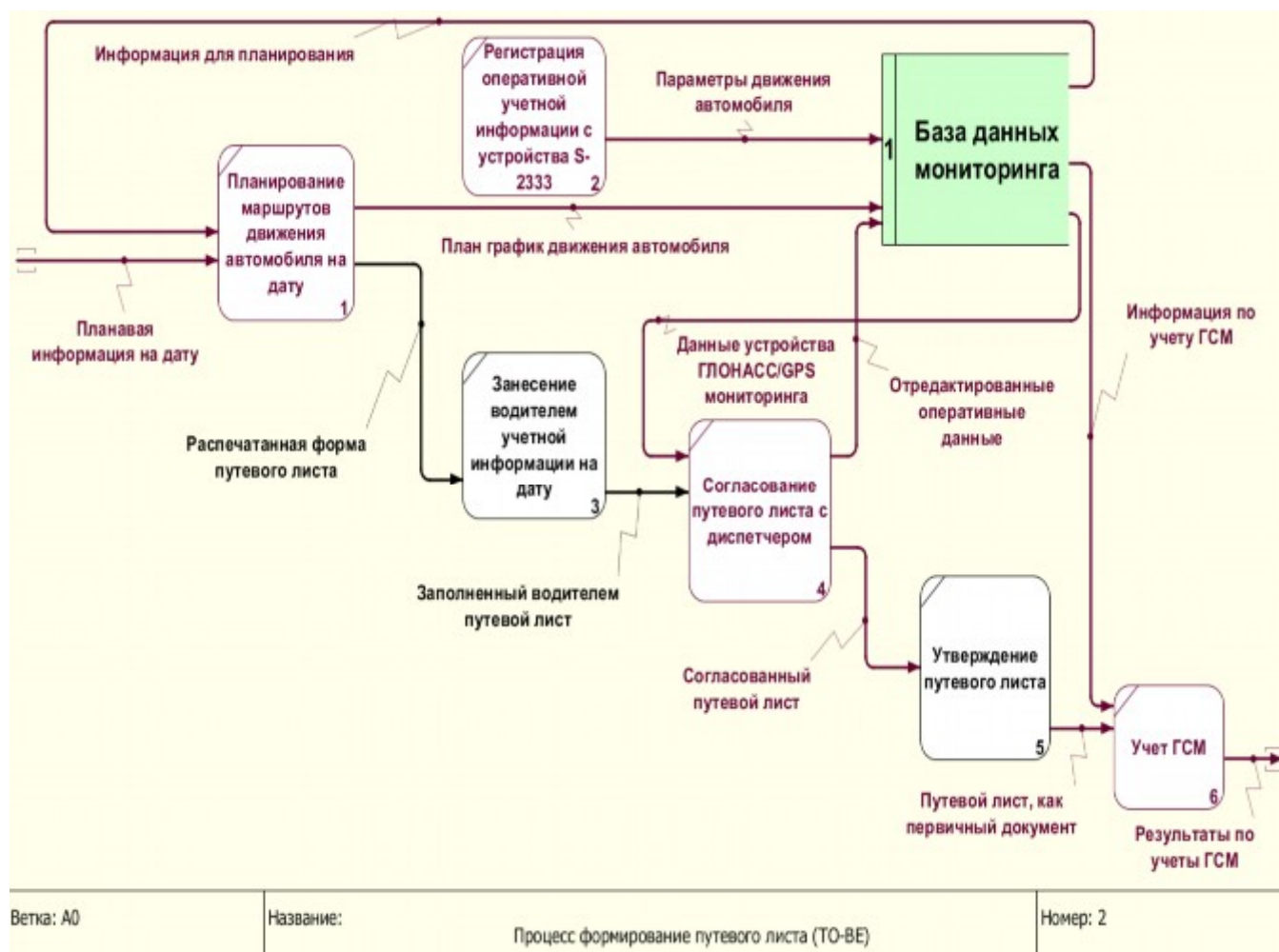


Рисунок 30 – Диаграмма декомпозиции процесса формирования путевого листа в модели TO-BE

Таким образом, использование простого и недорогого устройства мониторинга s-2333, связанного с оборудованием Глонас/GPS и имеющее весь необходимый функционал для реализации качественного мониторинга транспорта, позволяет формировать соответствующую БД и реализовать новую систему формирования путевого листа. Система обеспечивает повышение оперативности и достоверности первичных учетных данных о грузоперевозках и позволяет сократить непроизводительные затраты ГСМ.

Литература

- 1 Грекул В. И. Проектное управление в сфере информационных технологий [Электронный ресурс] / В. И. Грекул, Н. В. Коровкина, Ю. В. Куприянов. — Эл. изд. — М. : БИНОМ. Лаборатория знаний, 2013. — 336 с. : ил. — (Проекты, программы, портфели).
- 2 Ехлаков Ю.П. Введение в программную инженерию : учебное пособие / Ю.П. Ехлаков. — Томск: Эль Контент, 2011. — 148с.
Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=209001
- 3 ISO/IEC 25000:2014(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE [Электронный ресурс]. — Режим доступа: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25000:ed-2:v1:en>
- 4 ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного

обеспечения (SQuaRE). Модели качества систем и программных продуктов [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200121069>

6 Решетова Н.Э. Актуально ли на сегодня моделирование в IDEF0? / <http://projectimo.ru/biznes-processy/idef0.html>

– (Время успешных проектов) (обращение 10.09.2020 г.)

7 Маклаков С.В. BPWin и ERWin. CASE-средства разработки информационных систем. – М.: ДИАЛОГ-МИФИ, 2000.

8. Метелкин К.О., Астахова А.В. Особенности разработки серверного ПО для системы навигационного мониторинга грузовых перевозок : Горизонты образования, 2017, вып.19. Приложение – XIV Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь - 2017". Секция «Информационные технологии». Подсекция «Программная инженерия». / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2017. – С.26-29. <http://edu.secna.ru/publication/5/release/147/attachment/39/>

9 Ramus Educational: свободно распространяемое ПО / <http://www.ramussoftware.com>

10 Астахова А.В. Проектирование систем информации и управления: Учебник. – Барнаул: Изд-во АлтГТУ им. И.И. Ползунова, 2011.

11 Астахова, А.В., Ананьев, Т.П., Васильева, О.В., Данилин, А.В. Вопросы обеспечения качества программных продуктов в IT-проектах систем организационного управления. / А.В. Астахова, Т.П. Ананьев и др. // Горизонты образования, 2018, вып.20. Приложение – XV Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь – 2018". Секция «Информационные технологии». Подсекция «Программная инженерия». Номер гос. регистрации 0229905270, регистр. свидет.4972 от 07.04.1999 г. / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2018. – С. 64-72. — Режим доступа: <http://edu.secna.ru/media/f/pi2018v1.pdf>

12 Астахова А.В. Информационные системы в экономике и защита информации на предприятиях – участниках ВЭД : учебное пособие / А.В. Астахова. – СПб. : Троицкий мост, 2014. – 216 с.: ил.