

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

И.Е. Еремин, В.В. Еремина, О.В. Жилиндина

МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ

Учебное пособие

Благовещенск
Издательство АмГУ
2020

УДК 519.713
ББК 22.379
Е 70

*Печатается по решению
ученого совета университета*

Рецензент:

*Чалкина Н.А., кандидат педагогических наук, доцент
кафедры общей математики и информатики Амурского го-
сударственного университета, доцент.*

Еремин И.Е., Еремина В.В., Жилиндина О.В.

Моделирование бизнес-процессов / И.Е. Еремин, В.В.
Еремина, О.В. Жилиндина. – Благовещенск: Изд-во АмГУ,
2020. – 39 с.

Пособие предназначено для самостоятельной работы студентов направления подготовки 09.03.01. – Информатика и вычислительная техника, 09.03.02. – Информационные системы и 38.03.05. – Бизнес-информатика изучающих дисциплину «Моделирование бизнес-процессов». Пособие содержит методические материалы по построению диаграмм бизнес-процессов, таких как функциональная модель, внешний и внутренний документооборот и последовательности выполнения действий.

© Амурский государственный университет, 2020

© Еремин И.Е., Еремина В.В., Жилиндина О.В., 2020

ВВЕДЕНИЕ

Бизнес-процесс – это логичный, последовательный, взаимосвязанный набор мероприятий, который потребляет ресурсы, создаёт ценность и выдаёт результат. Моделирование бизнес-процессов – это эффективное средство поиска путей оптимизации деятельности компании, позволяющее определить, как компания работает в целом и как организована деятельность на каждом рабочем месте. Под методологией (нотацией) создания модели (описания) бизнес-процесса понимается совокупность способов, при помощи которых объекты реального мира и связи между ними представляются в виде модели. Для каждого объекта и связей характерны ряд параметров, или атрибутов, отражающих определённые характеристики реального объекта (номер объекта, название, описание, длительность выполнения (для функций), стоимость и др.).

Описание бизнес-процессов проводится с целью их дальнейшего анализа и реорганизации. Целью реорганизации может быть внедрение информационной системы, сокращение затрат, повышение качества обслуживания клиентов, создание должностных и рабочих инструкций и т.п., а детальное описание процессов само по себе не представляет ценности. Реинжиниринг бизнес-процессов (англ. Business process reengineering) – это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов для достижения максимальной эффективности производственно-хозяйственной и финансово-экономической деятельности, оформленное соответствующими организационно-распорядительными и нормативными документами.

Основу многих современных методологий моделирования бизнес-процессов составили методология SADT (Structured Analysis and Design Technique – метод структурного анализа и проектирования), семейство стандартов IDEF (Icam DEFinition, где Icam – это Integrated Computer-Aided Manufacturing).

В первой части пособия рассматривается процесс построения функциональных моделей. Во второй части пред-

ложена методика построения диаграмм потоков данных. В третьей части описано построение диаграмм последовательности выполнения действий

1. ПОСТРОЕНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

1.1. Понятие бизнес-процесса

Бизнес-процесс определяется как логически завершенный набор взаимосвязанных и взаимодействующих видов деятельности, поддерживающий деятельность организации и реализующий её политику, направленную на достижение поставленных целей.

Бизнес-модель определяется как формализованное (графическое, табличное, текстовое, символическое) описание бизнес-процессов, отражающее реально существующую или предполагаемую деятельность предприятия.

Моделирование бизнес-процессов включает следующие цели:

- обеспечение понимания структуры организации и динамики происходящих в ней процессов;
- обеспечение понимания текущих проблем организации и возможностей их решения;
- обеспечение единого восприятия заказчиками, пользователями и разработчиками целей и задач организации;
- создание основы для формирования требований к программному обеспечению, автоматизирующему бизнес-процессы организации.

Модель бизнес-процесса должна определять:

- процедуры (функции, работы), которые необходимо выполнить для получения заданного конечного результата;
- последовательность выполнения процедур;
- механизмы контроля и управления в рамках рассматриваемого бизнес-процесса;
- субъекты выполнения процедур процесса;
- входящие документы / информацию, используемые каждой процедурой процесса;
- исходящие документы / информацию, генерируемые процедурами процесса;
- ресурсы, требующиеся для выполнения каждой процедуры процесса;
- документацию / условия, регламентирующие выполнение процедуры;

— параметры, характеризующие выполнение процедур и процесса в целом.

1.2. Методы моделирования бизнес-процессов

В основе методов моделирования бизнес-процессов могут лежать как структурный, так и объектно-ориентированный подходы к моделированию. К ним относят:

— метод функционального моделирования SADT/IDEF0;

— метод моделирования процессов IDEF3;

— моделирование потоков данных DFD;

— нотация моделирования потоков работ BPMN;

— метод ARIS;

— метод моделирования, используемый в технологии Rational Unified Process.

Методология структурного анализа и проектирования (Structured Analysis and Design Technique, SADT) была создана в конце 60-х гг. XX в. Дугласом Россом. SADT нашла своё применение в области описания большого количества сложных искусственных систем из широкого спектра областей. В 1973 г. впервые при помощи SADT был реализован крупный аэрокосмический проект. В виде конечного продукта SADT появилась в 1975 г. К 1981 г. методология SADT использовалась более чем в 50 компаниях при работе над проектами, охватывавшими различные проблемные области, в том числе телефонные сети, аэрокосмическое производство, управление и контроль, учёт материально-технических ресурсов. Причина успешного и разнообразного её применения заключается в том, что SADT является полной методологией для создания описания систем, основанной на концепциях системного моделирования.

Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (ICAM DEFinition), являющегося основной частью программы ICAM (Integrated Computer Aided Manufacturing — интегрированная компьютеризация производства), проводимой по инициативе военно-воздушных сил США. Метод SADT реализован в одном из стандартов этого

семейства — IDEF0, который был утверждён в качестве федерального стандарта США в 1993 г.

Методологии семейства IDEF позволяют отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах.

В настоящий момент в семейство IDEF входят:

— IDEF0 — методология функционального моделирования (изучаемая система представляется в виде набора взаимосвязанных функций — функциональных блоков);

— IDEF1 — методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи;

— IDEF1X (IDEF1 eXtended) — методология построения реляционных структур (как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе);

— IDEF2 — методология динамического моделирования развития систем;

— IDEF3 — методология документирования процессов, происходящих в системе;

— IDEF4 — методология построения объектно-ориентированных систем, позволяющая наглядно отображать структуру объектов и заложенные принципы их взаимодействия;

— IDEF5 — методология онтологического исследования сложных систем при помощи определённого словаря терминов и правил, на основании которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени;

— IDEF6 — методология использования рационального опыта проектирования, позволяющая предотвратить возникновение структурных ошибок при новом проектировании информационных систем;

— IDEF7 — методология аудита информационной системы;

— IDEF8 — методология разработки модели графического интерфейса пользователя;

— IDEF9 — методология анализа существующих условий и ограничений, их влияния на принимаемые решения в процессе реинжиниринга;

— IDEF10 — методология моделирования архитектуры выполнения;

— IDEF11 — методология информационного моделирования артефактов;

— IDEF12 — методология организационного моделирования;

— IDEF13 — методология проектирования трёхсхемного дизайна карт;

— IDEF14 — методология моделирования компьютерных сетей.

1.3. Методология SADT/IDEF0

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

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

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

Методология SADT/IDEF0 представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отобра-

жает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Основные элементы методологии SADT/IDEF0 основываются на следующих концепциях:

- графическое представление блочного моделирования — графика блоков и дуг SADT-диаграммы — отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него;

- строгость и точность — выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика;

- связность диаграмм — блоки нумеруются специальным образом;

- уникальность меток и наименований — отсутствие повторяющихся имён;

- синтаксические правила для графических символов (блоков и стрелок);

- разделение входов и управлений — правило определения роли данных;

- отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель.

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

Одним из основных понятий стандарта IDEF0 является декомпозиция (Decomposition).

Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели. Декомпозиция позволяет постепенно

и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм.

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

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

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

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

Далее блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединённых интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть разбита подобным образом для более детального представления.

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

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

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

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

— ограничение количества функциональных блоков на каждом уровне декомпозиции тремя–шестью (верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать её создание);

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

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

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

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

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

Графическим отображением интерфейсной дуги является однонаправленная стрелка. Каждая интерфейсная дуга должна иметь своё уникальное наименование (Arrow Label). По требованию стандарта, наименование должно быть оборотом существительного. Началом и концом каждой функциональной дуги могут быть только функциональные блоки, при этом источником может быть только выходная сторона

блока, а приёмником — любая из трёх оставшихся. Каждый функциональный блок должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую.

Типизацию категорий информации можно описать аббревиатурой ICOM:

I (Input), вход — то, что потребляется в ходе выполнения процесса;

C (Control), управление — ограничения и инструкции, влияющие на выполнение процесса;

O (Output), выход — то, что является результатом выполнения процесса;

M (Mechanism), механизм — то, что используется для выполнения процесса, но остаётся неизменным (рис. 1).



Рисунок 1 – Стрелки на диаграмме IDEF0

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

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

Управление остаётся неизменным при работе блока. Если же некая инструкция или правило должно быть измене-

но блоком, то соответствующую информацию следует рассматривать не как управление, а как входные данные. В случае, когда неясно, относить стрелку к входу или к управлению, следует отнести её к управлению, вплоть до разрешения неясности.

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

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

Выделяют пять основных видов связей: по входу, по управлению, обратная по входу, обратная по управлению, выход-механизм.

Связь по входу применяется, когда один из блоков должен полностью завершить работу перед началом работы другого блока (рис.2).

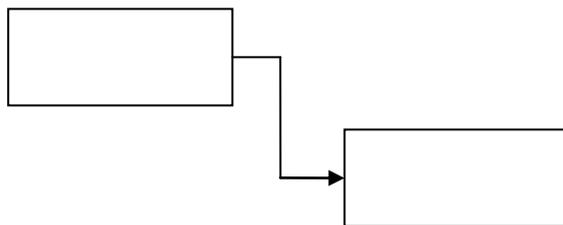


Рисунок 2. - Связь по входу

Связь по управлению показывает, что один блок управляет работой другого (рис. 3).

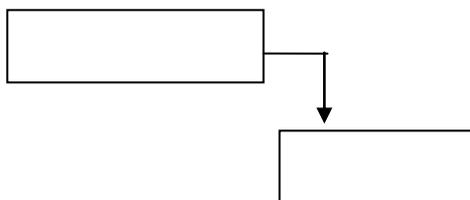


Рисунок 3. - Связь по управлению

Связь выход–механизм показывают, что выход одного функционального блока применяется в качестве инструментария для работы второго (рис. 4).

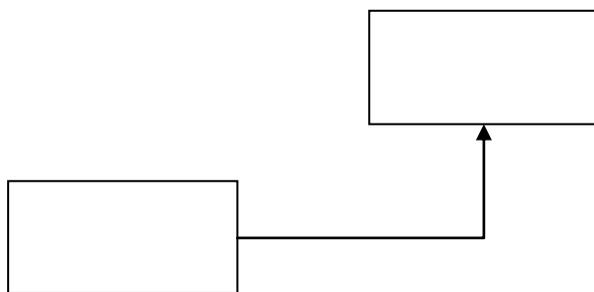


Рисунок 4. - Связь выход–механизм

Связь обратная по управлению применяется в случае, когда зависимый блок корректирует исполнение управляющего блока (рис. 5).

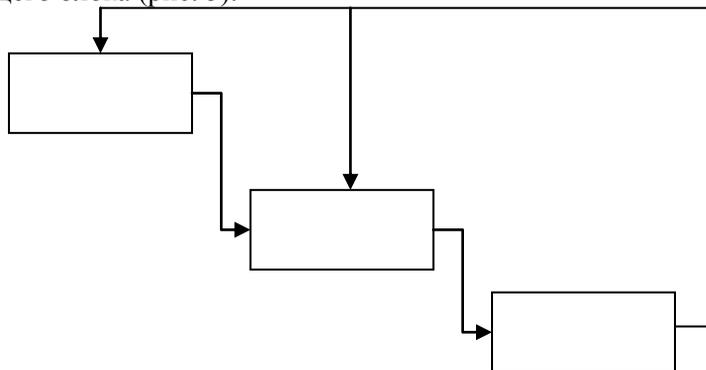


Рисунок 5. Связь обратная по управлению

Связь обратная по входу обычно применяется для описания циклов повторной обработки чего-либо (рис. 6). Кроме того, возможно применение данной связи при повторном использовании бракованной продукции.

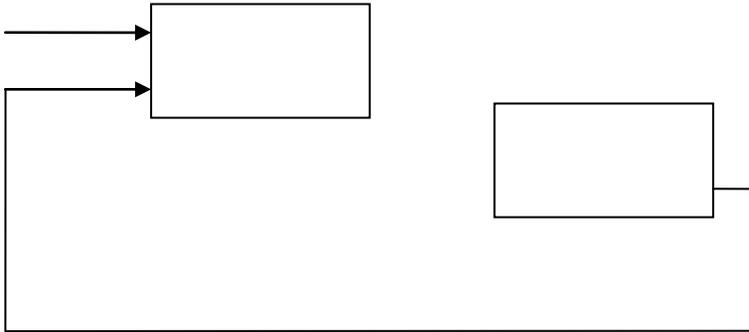


Рисунок 6. - Связь обратная по входу

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

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

1.4. Пример построения функциональной модели

Построение модели системы должно начинаться с изучения всех документов, описывающих ее функциональные возможности. Одним из таких документов является техническое задание, а именно разделы «Назначение разработки», «Цели и задачи системы» и «Функциональные характеристики системы».

После изучения исходных документов и опроса заказчиков и пользователей системы необходимо сформулировать цель моделирования и определить точку зрения на модель. Рассмотрим технологию ее построения на примере системы «Служба занятости в рамках вуза».

Сформулируем цель моделирования: описать функционирования системы, которое было бы понятно ее пользователю, не вдаваясь в подробности, связанные с реализацией. Модель будем строить с точки зрения пользователей (студент, преподаватель, администратор, деканат, фирма).

Начнем с построения контекстной IDEF0-диаграммы. Согласно описанию системы основной функцией является обслуживание ее клиентов посредством обработки запросов, от них поступающих. Таким образом, определим единственную работу контекстной диаграммы как «Обслужить клиента системы». Далее определим входные и выходные данные, а также механизмы и управление.

Для того чтобы обслужить клиента, необходимо зарегистрировать его в системе, открыть доступ к БД и обработать его запрос. В качестве входных данных будут использоваться «имя клиента», «пароль клиента», «исходная БД», «запрос клиента». Выполнение запроса ведет либо к получению информации от системы, либо к изменению содержимого БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться «отчеты» и «измененная БД». Процесс обработки запросов будет выполняться монитором системы под контролем администратора.

Таким образом, определим контекстную диаграмму системы (рис. 7).

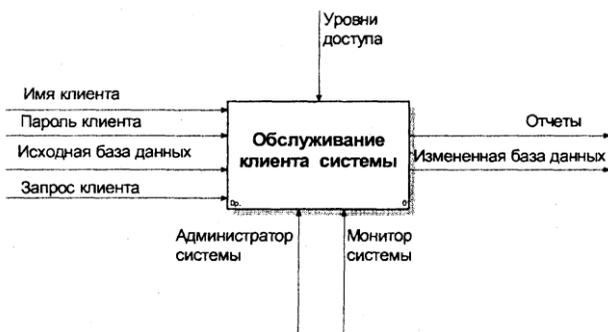


Рисунок 7. – Контекстная диаграмма

Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

1. Определение уровня доступа в систему.
2. Выбор подсистемы.
3. Обращение к подсистеме.
4. Изменение БД (при необходимости).

Получим диаграмму, изображенную на рис. 8.



Рисунок 8. – Декомпозиция контекстной диаграммы

Закончив декомпозицию контекстной диаграммы, переходят к декомпозиции диаграммы следующего уровня. Обычно при рассмотрении третьего и более нижних уровней модели возвращаются к родительским диаграммам и корректируют их.

Декомпозируем последовательно все блоки полученной диаграммы. Первым этапом при определении уровня доступа в систему является определение категории пользователя. По имени клиента осуществляется поиск в базе пользователей, определяя его категорию. Согласно определенной категории выясняются полномочия, предоставляемые пользователю системы.

Далее проводится процедура доступа в систему, проверяя имя и пароль доступа. Объединяя информацию о полномочиях и уровне доступа в систему, для пользователя формируется набор разрешенных действий.

Таким образом, определение уровня доступа в систему будет выглядеть как показано на рис. 9.

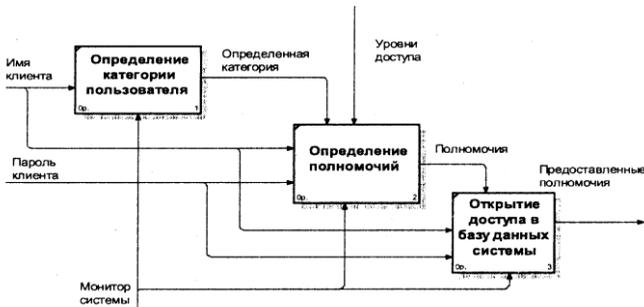


Рисунок 9. – Декомпозиция работы «Определение уровня доступа»

После прохождения процедуры доступа в систему монитор анализирует запрос клиента, выбирая подсистему, которая будет обрабатывать запрос.

Декомпозиция работы «Обращение к подсистеме» не отвечает цели и точке зрения модели. Пользователя системы не интересуют внутренние алгоритмы ее работы. В данном случае ему важно, что выбор подсистемы будет произведен автоматически, без его вмешательства, поэтому декомпозиция обращения к подсистеме только усложнит модель.

Контрольные вопросы

1. Что такое бизнес-процесс

2. Какие цели включает моделирование бизнес-процессов
3. Что определяет модель бизнес-процесса.
4. Какие методы моделирования бизнес-процессов существуют
5. Какие методологии входят в семейство IDEF
6. Что представляет собой модель в нотации IDEF0
7. Что обозначают работы в IDEF0
8. Перечислите типы стрелок.
9. Назовите виды взаимосвязей.

2. ПОСТРОЕНИЕ ДИАГРАММ ПОТОКОВ ДАННЫХ

2.1. Методология DFD

Нотация диаграмм потоков данных позволяет отображать на диаграмме как шаги бизнес-процесса, так и поток документов и управления (в основном управления, поскольку на верхнем уровне описания процессных областей значение имеет передача управления). Также на диаграмме можно отображать средства автоматизации шагов бизнес-процессов.

Области применения диаграмм потоков данных:

- моделирование функциональных требований к проектируемой системе;

- моделирование существующего процесса движения информации;

- описание документооборота, обработки информации;

- дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота;

- проведение анализа и определения основных направлений реинжиниринга информационной системы.

Диаграммы DFD могут дополнить то, что уже отражено в модели IDEF0, поскольку они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией как внутри системы между бизнес-функциями, так и системы в целом с внешней информационной средой.

Методика DFD удобна для описания не только бизнес-процессов (как дополнение к IDEF0), но и программных систем:

- DFD-диаграммы создавались как средство проектирования программных систем (в то время как IDEF0 — средство проектирования систем вообще), поэтому DFD имеют более богатый набор элементов, адекватно отражающих их специфику (например, хранилища данных являются образами файлов или баз данных);

- наличие мини-спецификаций DFD-процессов нижнего уровня позволяет преодолеть логическую незавершенность IDEF0 (моделирование обрывается на некотором достаточно низком уровне, когда дальнейшая детализация моде-

ли становится бессмысленной) и построить полную функциональную спецификацию разрабатываемой системы.

С помощью DFD-диаграмм требования к проектируемой информационной системе разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель декомпозиции DFD-функций — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

На схемах бизнес-процесса отображаются:

- функции процесса;
- входящая и исходящая информация при описании документов;
- внешние бизнес-процессы, описанные на других диаграммах;
- точки разрыва при переходе процесса на другие страницы.

При моделировании DFD система рассматривается как сеть связанных между собой функций, т. е. как совокупность сущностей. Методология основана на идее нисходящей иерархической организации. Целью является преобразование общих, неясных знаний о требованиях к системе в точные определения. Внимание фокусируется на потоках данных. Методология поддерживается традиционными нисходящими методами проектирования.

2.2. Варианты методологии DFD

Существует два основных варианта методологии DFD: методология Гейна–Сарсона (Gane–Sarson) и методология Йордана–Де Марко (Yourdon–DeMarko).

Главной отличительной чертой методологии Гейна–Сарсона является наличие этапа моделирования данных, определяющего содержимое хранилищ данных (БД и файлов) в DFD. Этот этап включает построение списка элементов данных, располагающихся в каждом хранилище данных; анализ отношений между данными и построение соответствующей диаграммы связей между элементами данных; представление всей информации по модели в виде связанных нормализованных таблиц.

Кроме того, эти методологии отличаются нотацией.

Обе методологии основаны на простой концепции нисходящего поэтапного разбиения функций системы на подфункции:

1) формирование контекстной диаграммы верхнего уровня, идентифицирующей границы системы и определяющей интерфейсы между системой и окружением;

2) формирование списка внешних событий, на которые система должна реагировать (после интервьюирования эксперта предметной области), для каждого из которых строится пустой процесс (Bubble) в предположении, что его функция обеспечивает требуемую реакцию на это событие (в большинстве случаев включает генерацию выходных потоков и событий);

3) проведение детализации для каждого из пустых процессов.

Методология DFD позволяет уже на стадии функционального моделирования определить базовые требования к данным. В этом случае совместно используются методологии DFD и IDEF1X.

2.3. Синтаксис и семантика моделей DFD

Диаграммы потоков данных применяются для графического представления (Flowchart) движения и обработки информации. Обычно диаграммы этого типа используются для проведения анализа организации информационных потоков и для разработки информационных систем. Каждый блок в DFD может развёртываться в диаграмму нижнего уровня, что позволяет на любом уровне абстрагироваться от деталей.

DFD-диаграммы отображают внешний и внутренний документооборот. Внешний документооборот – это отображение потока документов между предприятием и внешними по отношению к нему сущностями. Внутренний документооборот это движение документов внутри предприятия, т.е. между отделами.

Основные объекты нотации DFD:

— блоки (Blocks) или работы (Activities) — отображают процессы обработки и изменения информации или отделы предприятия;

— стрелки (Arrows) или потоки данных (Data Flow) — отображают информационные потоки;

— хранилища данных (Data Store) — отображают данные, к которым осуществляется доступ; эти данные используются, создаются или изменяются работами;

— внешние ссылки (External References) или внешние сущности (External Entity) — отображают объекты, с которыми происходит взаимодействие.

Работы DFD — графическое изображение операции по обработке или преобразованию информации (данных). Входная информация преобразуется в выходную. Работы именуются глаголами в неопределённой форме с последующим

По нотации Гейна–Сарсона DFD-блок изображается прямоугольником со скруглёнными углами. Каждый блок должен иметь уникальный номер для ссылки на него внутри диаграммы.

Во избежание пересечений линий потоков данных один и тот же элемент может на одной и той же диаграмме отображаться несколько раз; в таком случае два или более прямоугольников, обозначающих один и тот же элемент, могут идентифицироваться линией, перечёркивающей нижний правый угол.

Поток данных — механизм, использующийся для моделирования передачи информации между участниками процесса информационного обмена (отделами, хранилищами данных, внешними ссылками). По нотации Гейна–Сарсона поток данных изображается стрелкой между двумя объектами DFD-диаграммы, предпочтительно горизонтальной и/или вертикальной, причём направление стрелки указывает направление потока. Каждая стрелка должна иметь источник и цель. В отличие от стрелок IDEF0-диаграммы, стрелки DFD могут входить или выходить из любой стороны блока.

В DFD-диаграммах для описания диалогов типа команды-ответа между операциями применяются двунаправленные стрелки между функцией и внешней сущностью и/или между внешними сущностями. Стрелки могут сливаться и разветвляться, что позволяет описать декомпозицию стрелок. Каж-

дый новый сегмент сливающейся или разветвляющейся стрелки может иметь собственное имя.

Иногда информация может двигаться в одном направлении, обрабатываться и возвращаться обратно. Такая ситуация может моделироваться либо двумя различными потоками, либо одним двунаправленным потоком. На поток данных можно ссылаться, указывая процессы, сущности или накопители данных, которые поток соединяет. Каждый поток должен иметь имя, расположенное вдоль или над стрелкой, выбранное таким образом, чтобы в наибольшей степени передавать смысл содержания потока пользователям, которые будут рассматривать диаграмму потоков данных.

Хранилище данных — графическое представление потоков данных, импортируемых/экспортируемых из соответствующих баз данных, изображает объекты в покое. Является неким прообразом базы данных информационной системы организации.

Хранилища данных предназначены для изображения неких абстрактных устройств для хранения информации, которую можно в любой момент времени поместить или извлечь из них, безотносительно к их конкретной физической реализации.

Хранилища данных используются:

— в материальных системах (там, где объекты ожидают обработки, например в очереди);

— в системах обработки информации для моделирования механизмов сохранения данных с целью дальнейших операций.

По нотации Гейна—Сарсона хранилище данных обозначается двумя горизонтальными линиями, замкнутыми с одного края.

В модели может быть создано множество вхождений хранилищ данных, каждое из которых может иметь одинаковое имя и ссылочный номер. Для того чтобы не усложнять диаграмму потоков данных пересечениями линий, можно изображать дубликаты накопителя данных дополнительными вертикальными линиями с левой стороны квадрата. Храни-

лица данных не изменяют потоки данных, а служат только для хранения поступающих объектов.

Внешняя сущность — объект диаграммы потоков данных, являющийся источником или приёмником информации извне модели. Внешние сущности изображают входы и/или выходы, т. е. обеспечивают интерфейс с внешними объектами, находящимися вне моделируемой системы. Внешними ссылками системы обычно являются логические классы предметов или людей, представляющие собой источник или приёмник сообщений, например, заказчики, конструкторы, технологи, производственные службы, поставщики и т. д.

По нотации Гейна–Сарсона пиктограмма внешней ссылки представляет собой отенённый прямоугольник, верхняя левая сторона которого имеет двойную толщину и обычно располагается на границах диаграммы.

Одна и та же внешняя ссылка может быть использована многократно на одной или нескольких диаграммах. Обычно к такому приёму прибегают для того, чтобы не рисовать слишком длинных и запутанных стрелок. Каждая внешняя сущность имеет префикс.

При рассмотрении системы как внешней функции часто указывается, что она находится за пределами границ моделируемой системы. Преобразования потоков данных во внешних сущностях игнорируются.

Помимо этого, для каждого информационного потока и хранилища определяются связанные с ними элементы данных. Каждому элементу данных присваивается имя. Также для него может быть указан тип данных и формат. Именно эта информация является исходной на следующем этапе проектирования — построении модели «сущность–связь». При этом, как правило, информационные хранилища преобразуются в сущности. Проектировщику остаётся только решить вопрос с использованием элементов данных, не связанных с хранилищами.

2.4. Пример построения внешнего и внутреннего документооборота

В бухгалтерии ООО ЧОО «Феликс» обрабатываются следующие документы:

- бухгалтерский баланс;
- отчеты о прибылях и убытках;
- первичные документы (декларации по налогам, платежные поручения, кассовые ордера, выписка банка, банковские гарантии, справки о состоянии расчетного счета и др.);
- данные налогового и управленческого учета.

Взаимодействие ООО ЧОО «Феликс» с отделением УФНС по Амурской области происходит следующим образом: предприятие передает отчеты, а отделение УФНС по Амурской области передает нормативные документы, проводит проверку правильной уплаты налогов.

ООО ЧОО «Феликс» взаимодействует с отделением УПФ РФ по Амурской области, получая заявку на денежную сумму, а затем отправляет платежное поручение; с налоговой службой, которая направляет запрос с требованием предоставить налоговую декларацию и через некоторое время предприятие должно предоставить необходимые документы.

ООО ЧОО «Феликс» также взаимодействует с ПАО «Сбербанк России». В банк отправляются данные о счетах компании и выполняются платежные поручения. Все работники получают заработную плату на карты этого банка.

ООО ЧОО «Феликс» в случае необходимости взаимодействует с поставщиками. Предприятие делает заказ на закупку необходимых товаров и услуг, подписывается договор. Производится накладная о товарах, выписываются счета за услуги

Взаимодействие предприятия с клиентами заключается в осуществлении реализации услуг. Клиент делает запрос на предоставление услуги, а предприятие предоставляет договор.

Итоговая диаграмма внешнего документооборота представлена на рисунке 10.

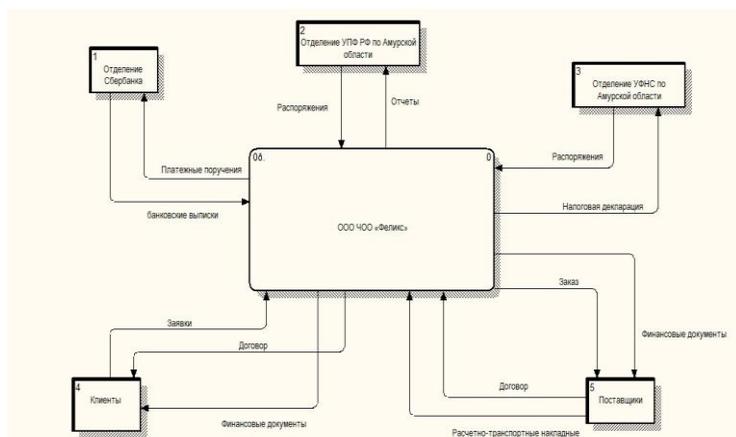


Рисунок 10 – Внешний документооборот

Функции, выполняемые участниками внутреннего документооборота:

- бухгалтер занимается финансовой деятельностью предприятия, а также ведёт взаимодействие с налоговой инспекцией, пенсионным фондом и банком, предоставляя все обязательные виды отчетности. Внутри предприятия получает приказы от директора и передает ему отчеты, а также получает таблицу рабочего времени и сведения о сотрудниках от начальника отдела кадров;

- генеральный директор отдает приказы бухгалтерии, начальнику отдела кадров и заместителю директора так же получает отчеты от них;

- заместитель директора принимает заявки на услуги, отдает договора клиентам, а внутри предприятия отдает приказы охранникам, и специалистам по установке сигнализации, а после получает от них отчеты;

- начальник отдела кадров передает сведения о сотрудниках и таблицу рабочего времени в бухгалтерию, получает приказы от генерального директора, а после передает отчеты генеральному директору;

- охранники получают приказы от заместителя директора и передают ему отчеты;

– специалисты по установке сигнализации передают заказ, если была заявка на установку сигнализации, а внутри предприятия получают приказы от директора и передают ему отчеты.

11. Внутренний документооборот представлен на рисунке

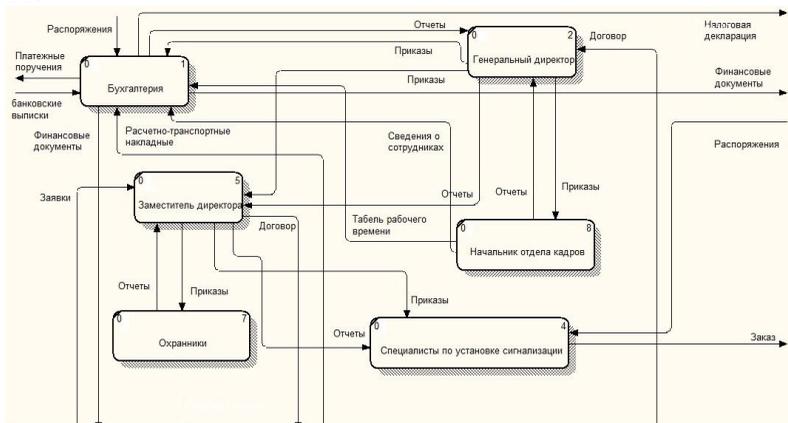


Рисунок 11 – Внутренний документооборот

Контрольные вопросы

1. Области применения диаграмм потоков данных
2. Что отображается на схемах бизнес-процесса
3. Какие варианты методологии DFD существуют
4. На какой концепции основаны методологии DFD
5. Работа на диаграмме DFD
6. Поток данных на диаграмме DFD
7. Хранилища данных на диаграмме DFD
8. Внешние ссылки на диаграмме DFD

3. ПОСТРОЕНИЕ ДИАГРАММЫ ПОСЛЕДОВАТЕЛЬНОСТИ ВЫПОЛНЕНИЯ ДЕЙСТВИЙ

3.1. Методика IDEF3. Модель процессов

IDEF3 предлагает структурный метод описания процессов. Модель описывается как упорядоченная последовательность событий. Методика IDEF3 хорошо приспособлена для сбора данных.

Данная методика не имеет жёстких синтаксических и семантических ограничений. Очень часто IDEF3 используют как метод, дополняющий IDEF0.

Каждый функциональный блок IDEF0 может быть представлен в виде отдельного процесса IDEF3.

Основой методологии является сценарий (Scenario) бизнес-процесса, осуществляющий описание последовательности изменений свойств объекта в рамках рассматриваемого процесса. Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков — документов, определяющих структуру и последовательность процесса, и документов, отображающих ход его выполнения.

В IDEF3 используется два типа диаграмм, представляющие описание одного и того же сценария в разных ракурсах.

1) С помощью диаграмм описания последовательности этапов процесса (Process Flow Description Diagrams, PFDD) документируется последовательность и описание стадий обработки в рамках исследуемого бизнес-процесса. Описание производится с точки зрения стороннего наблюдателя. Ключевыми элементами являются понятия, процесс, логика процесса.

2) Диаграммы перехода состояния объекта (Object State Transition Network, OSTN) используются для иллюстрации трансформаций, которые происходят на каждой стадии бизнес-процесса. При этом описание производится с точки зрения самого объекта.

Действие в IDEF3 называется единицей работы (Unit of Work, UOW) и обозначается прямоугольником. Действия

именуются глаголами или отглагольными существительными.

С помощью связей выделяются существенные взаимоотношения между действиями, задавая их последовательность. Все связи однонаправленные.

Обычно стрелки рисуют слева направо, выходящими из правой и входящими в левую сторону блоков, либо сверху вниз (что является лишь соглашением, а не нормируется).

Выделяют три вида связей:

— связь Временное предшествование показывает, что исходное действие должно завершиться прежде, чем начнется выполнение конечного действия, а также, что исходное действие может инициировать выполнение конечного действия;

— связь Объектный поток применяется в случае, когда объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия;

— связь Нечёткое отношение используется, если невозможно применить предыдущие типы связей.

При помощи связи типа Нечёткое отношение можно, например, описывать параллельные процессы. Корректная интерпретация нечётких связей должна быть документирована в модели.

Одно действие может порождать несколько. Или для выполнения действия требуется завершение нескольких действий. Для описания ветвлений процессов используют перекрестки (таблица 1).

Таблица 1 – Типы перекрестков

Наименование	Смысл в случае слияния	Смысл в случае разветвления
Asynchronous	Все предшествующие должны быть закончены.	Все следующие процессы должны быть запущены .
Synchronous AND	Все предшествующие процессы завершаются одновременно.	Все следующие процессы запускаются одновременно.

Наименование	Смысл в случае слияния	Смысл в случае разветвления
Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены.	Один или несколько следующих процессов должны быть запущены.
Synchronous OR	Один или несколько предшествующих процессов завершаются одновременно.	Один или несколько следующих процессов запускаются одновременно.
XOR (Exclusive OR)	Только один процесс завершения.	Только один следующий процесс запускается.

Объекты – ссылки являются специальными символами, которые ссылаются на внешние части описания процесса. Они добавляются на диаграмму для того, чтобы обратить внимание редактора на что-либо важное, что невозможно связать со стрелкой, работой или перекрестком.

Объект-ссылки отображается в виде прямоугольника. Объекты-ссылки должны быть связаны с единицами работ или перекрестками пунктирными линиями.

При внесении объектов-ссылок необходимо указать их тип (таблица 2).

Таблица 2 – Типы объектов ссылок

Тип объекта	Цель описания
ОБЪЕСТ	Описывает участие важного объекта в работе
GOTO	Инструмент циклического перехода (в повторяющейся последовательности работ), возможно на текущей диаграмме, но не обязательно. Если все работы цикла присутствуют на текущей диаграмме, цикл может также изображаться стрелкой, возвращающейся на стартовую работу. GOTO может ссылаться на перекресток

Тип объекта-	Цель описания
UOB (Unit of behavior)	Применяется, когда необходимо подчеркнуть множественное использование какой-либо работы, но без цикла. Например. «Контроль качества» может быть использована, в процессе «Изготовления изделия» несколько раз ». После каждой единичной операции. Обычно 'этот тип ссылки не используется для моделирования автоматизированных операций.
NOTE	Используется для документирования важной информации, относящийся каким – либо графическим объектам на диаграмме. NOTE является альтернативой внесению текстового объекта в диаграмму.

3.3. Пример построения диаграммы последовательности выполнения действий

Проведем декомпозицию работы «Сборка настольных компьютеров» на диаграмму IDEF3. Данная работа начинает выполняться, когда поступают заказы на сборку. Первым действием проверяется наличие необходимых для сборки комплектующих и заказ со склада отсутствующих. Далее комплектующие подготавливаются для последующей сборки (освобождение от упаковки, снятие заглушек и т.п.).

Следующим шагом начинается непосредственно сам процесс сборки: установка материнской платы в корпус и процессора на материнскую плату, установка ОЗУ и винчестера. Данные действия выполняются всегда, независимо от конфигурации компьютера. Далее по желанию клиента могут быть установлены некоторые дополнительные комплектующие - DVD привод, ТВ-тюнер, кард-ридер. На этом сборка компьютера завершается. Следующим шагом идет установка операционной системы. По желанию клиента также может быть установлено дополнительное программное обеспечение. Последним действием составляется отчет о проделанной работе.

Диаграмма последовательности выполнения действий представлена на рисунке 12.

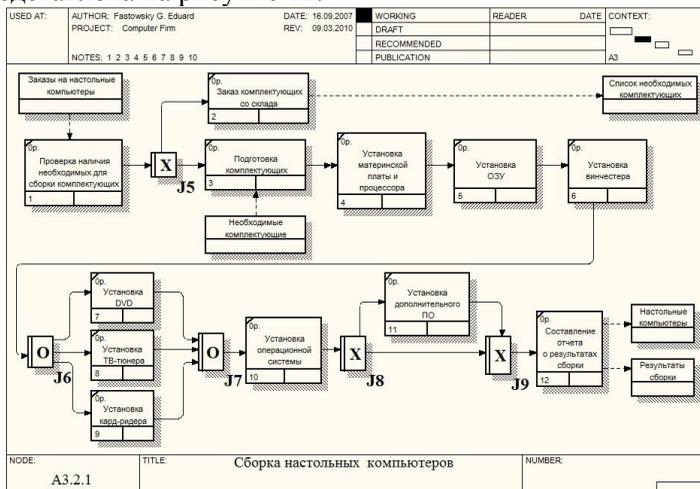


Рисунок 12 - Диаграмма последовательности выполнения действий

Рассмотрим основные особенности этой диаграммы. После проверки наличия необходимых для сборки комплектующих возможно одно из двух действий - или заказ со склада недостающих комплектующих, или, если все комплектующие в наличии, их подготовка. Поэтому необходимо поставить перекресток разветвления типа «Исключающее ИЛИ». Работы «Подготовка комплектующих» и «Установка материнской платы и процессора» соединены связью «Поток объектов». Тем самым показано, что между этими работами передаются объекты. Все последующие работы соединяются связями «старшая стрелка», поскольку они только показывают последовательность действий над одними и теми же объектами.

После установки винчестера возможна установка DVD привода, ТВ-тюнера, кард-ридера или любая их комбинация. Поэтому поставлен перекресток разветвления типа «Асинхронное ИЛИ». Такой же перекресток стоит и после завершения этих работ. Далее после установки операционной системы может быть установлено дополнительное ПО, или же

сразу формируется отчет, поэтому поставлен перекресток разветвления типа «Исключающее ИЛИ». За перекрестком разветвления типа «Исключающее ИЛИ» может следовать только такой же перекресток слияния, поэтому перед работой «Составление отчета о результатах сборки» поставлен такой же.

Контрольные вопросы

1. Что представляет собой методика IDEF3
2. Что является основой методологии IDEF3
3. Какие типы диаграмм используются в IDEF3
4. Виды связей
5. Типы перекрестков
6. Типы объектов ссылок

СПИСОК ЛИТЕРАТУРЫ

1. Долганова, О. И. Моделирование бизнес-процессов : учебник и практикум для академического бакалавриата / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова ; под ред. О. И. Долгановой. — М. : Издательство Юрайт, 2018. — 289 с.

2. Каменнова, М. С. Моделирование бизнес-процессов. В 2 ч. Часть 1 : учебник и практикум для бакалавриата и магистратуры / М. С. Каменнова, В. В. Крохин, И. В. Машков. — М. : Издательство Юрайт, 2018. — 282 с.

3. Каменнова, М. С. Моделирование бизнес-процессов. В 2 ч. Часть 2 : учебник и практикум для бакалавриата и магистратуры / М. С. Каменнова, В. В. Крохин, И. В. Машков. — М. : Издательство Юрайт, 2018. — 228 с.

4. Куприянов, Ю. В. Модели и методы диагностики состояния бизнес-систем : учебное пособие для вузов / Ю. В. Куприянов, Е. А. Кутлуниин. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2018. — 128 с.

5. Моделирование систем и процессов. Практикум : учебное пособие для академического бакалавриата / В. Н. Волкова [и др.] ; под ред. В. Н. Волковой. — М. : Издательство Юрайт, 2018. — 295 с.

6. Выюненко, Л. Ф. Имитационное моделирование : учебник и практикум для академического бакалавриата / Л. Ф. Выюненко, М. В. Михайлов, Т. Н. Первозванская ; под ред. Л. Ф. Выюненко. — М. : Издательство Юрайт, 2018. — 283 с.

7. Умнова Е.Г. Моделирование бизнес-процессов с применением нотации BPMN [Электронный ресурс] : учебно-методическое пособие / Е.Г. Умнова. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2017. — 48 с.

8. Брезгин В.И. Моделирование бизнес-процессов с AllFusion Process Modeler 4.1. Часть 1 [Электронный ресурс] : рабочая тетрадь / В.И. Брезгин. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2015. — 80 с.

9. Брезгин В.И. Моделирование бизнес-процессов с AllFusion Process Modeler 4.1. Часть 2 [Электронный ресурс] :

лабораторный практикум / В.И. Брезгин. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2015. — 52 с.

10. Моделирование систем и процессов : учебник для академического бакалавриата / В. Н. Волкова [и др.] ; под ред. В. Н. Волковой, В. Н. Козлова. — М. : Издательство Юрайт, 2018. — 450 с.

11. Мамонова, В. Г. Моделирование бизнес-процессов : учебное пособие / В. Г. Мамонова, Н. Д. Ганелина, Н. В. Мамонова. — Новосибирск : Новосибирский государственный технический университет, 2012. — 43 с.

12. Цуканова, О. А. Методология и инструментарий моделирования бизнес-процессов : учебное пособие / О. А. Цуканова. — Санкт-Петербург : Университет ИТМО, 2015. — 101 с.

13. Мамонова, В. Г. Управление процессами. Часть 1. Подготовка бизнес-процессов к моделированию. Инструменты моделирования : учебное пособие / В. Г. Мамонова, И. Н. Томилов, Н. В. Мамонова. — Новосибирск : Новосибирский государственный технический университет, 2014. — 96 с.

14. Силич, В. А. Моделирование и анализ бизнес-процессов : учебное пособие / В. А. Силич, М. П. Силич. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2011. — 212 с.

ОГЛАВЛЕНИЕ

Введение.....	3
1. Построение функциональной модели.....	5
1.1. Понятие бизнес-процесса	5
1.2. Методы моделирования бизнес-процессов.....	6
1.3. Методология SADT/IDEF0.....	8
1.4. Пример построения функциональной модели.....	16
2. Построение диаграмм потоков данных	21
2.1. Методология DFD	21
2.2. Варианты методологии DFD	22
2.3. Синтаксис и семантика моделей DFD.....	23
2.4. Пример построения внешнего и внутреннего документооборота.....	26
3. Построение диаграммы последовательности выполнения действий.....	30
3.1. Методика IDEF3. Модель процессов.....	30
3.2. Пример построения диаграммы последовательности выполнения действий	33

Илья Евгеньевич Еремин,
*профессор кафедры информационных и управляющих
систем АмГУ, д-р техн. наук;*

Виктория Владимировна Еремина,
*доцент кафедры информационных и управляющих
систем АмГУ, канд. физ.-мат. наук*

Ольга Викторовна Жилиндина,
*доцент кафедры информационных и управляющих
систем АмГУ, канд. техн. наук*

Моделирование бизнес-процессов. Учебное пособие.

Изд-во АмГУ. Подписано к печати

Отпечатано в типографии АмГУ.