*Занятие № 4*

*Номер учебной группы:* П-16

*Фамилия, инициалы учащегося:* Язубец Е.В.

*Дата выполнения работы:* 11.11.2022

*Тема работы:* «Функциональное моделирование с использованием пакета All Fusion Process Modeler»

*Цель работы:* Создание декомпозиционных диаграмм

*Задание:* Разработать функциональную диаграмму на разрабатываемое ТЗ. Создать декомпозицию функциональной диаграммы. Ответить на вопросы.

*Результат выполнения работы*

**Задание 1**

Ответы на вопросы:

1. Что собой представляет методология IDEF0?

**Ответ:**

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

**Модель может содержать 4 типа диаграмм:**

1.Контекстная диаграмма (в каждой модели может быть только одна контекстная диаграмма);

2.Диаграмма декомпозиции;

3.Диаграмма дерева узлов;

4Диаграмма только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействие с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которая описывает каждый фрагмент и взаимодействие фрагментов, называется диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и т.д., до достижения нужного уровня подробности описания

1. Что собой представляет методология DFD?

**Ответ:**

Целью методологии является построение модели рассматриваемой сис*темы в виде диаграммы потоков данных (Data Flow Diagram – DFD). Диа*граммы потоков данных предназначены прежде всего для описания *документооборота и обработки информации, хотя допускают и представление других объектов.*

При создании диаграммы потоков данных используются *четыре основных понятия:*

–потоки данных,

–процессы (работы) преобразования входных потоков данных в выходные,

–внешние сущности,

–накопители данных (хранилища).

*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.

*Процессы*(работы) служат для преобразования входных потоков данных в выходные. 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*моделирует данные, которые будут сохраняться в памяти между процессами. Информация, которую содержит хранилище, 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 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 миниспецификации.

*Словари данных*являются каталогами всех элементов данных, присутствующих в DFD, включая потоки данных, хранилища и процессы, а также все их атрибуты. *Миниспецификации обработки*– описывают DFD-процессы нижнего уровня. Фактически миниспецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы.

1. Что собой представляет методология IDEF3?

**Ответ:**

В IDEF3 декомпозиция используется для детализации работ. Ме­тодология IDEF3 позволяет декомпозировать работу многократно, т.е. работа может иметь множество дочерних работ. Это позволяет в одной модели описать альтернативные потоки. Возможность множественной декомпозиции предъявляет дополнительные требования к нумерации работ. Так, номер работы состоит из номера родительской работы, версии декомпозиции и собственного номера работы на текущей диаграмме.

Отличием стандарта IDEF3 от классической методологии WFD является также использование на схеме бизнес-процесса элемента «объект ссылки», который связывается с работами и перекрестками. С помощью объектов ссылки показывается прочая важная информация, которую целесообразно зафиксировать при описании бизнес-процесса.

IDEF3 предполагает построение двух типов моделей:

1) модель, отражающая некоторые процессы в их логической последовательности, позволяющая увидеть, как функционирует предприятие;

2) модель, показывающая «сеть переходных состояний объекта», предлагающая вниманию аналитика последовательность состояний, в которых может оказаться объект при прохождении через определенный процесс.

С помощью диаграмм IDEF3 можно анализировать сценарии из реальной жизни, например, как закрывать магазин в экстренных случаях или какие действия должны выполнить менеджер и продавец при закрытии. Каждый такой сценарий содержит в себе описание процесса и может быть использован, чтобы наглядно показать или лучше задокументировать бизнес-функции предприятия.

1. Что вы понимаете под декомпозицией?

**Ответ:**

Декомпозиция — научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых.

Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части.

Декомпозиция системы чаще всего представляется в виде иерархического дерева, вершина которого – сама система, а уровни – выделенные подсистемы.

Подобные иерархические деревья для приложений можно строить с целью:

* сбора и сохранения информации о структуре приложения. Глубина расчленения (количество уровней) будет варьироваться в зависимости от того, кто будет пользоваться этой структурой и как.
* создания чек листа, который можно будет использовать в процессе тестирования. Для этого необходимо достичь очень глубокого расчленения.

1. Опишите последовательность создания декомпозиции функциональных диаграмм.

**Ответ:**

Первый способ – выделить декомпозируемую активность, кликнув на ней мышкой (или выделив блок в навигаторе модели на вкладке Activities), нажать кнопку Go to Child Diagram на панели инструментов

Второй способ – кликнуть правой клавишей мыши в навигаторе модели на вкладке Activities на блоке, который необходимо декомпозировать и из выпадающего списка выбрать пункт Decompose.

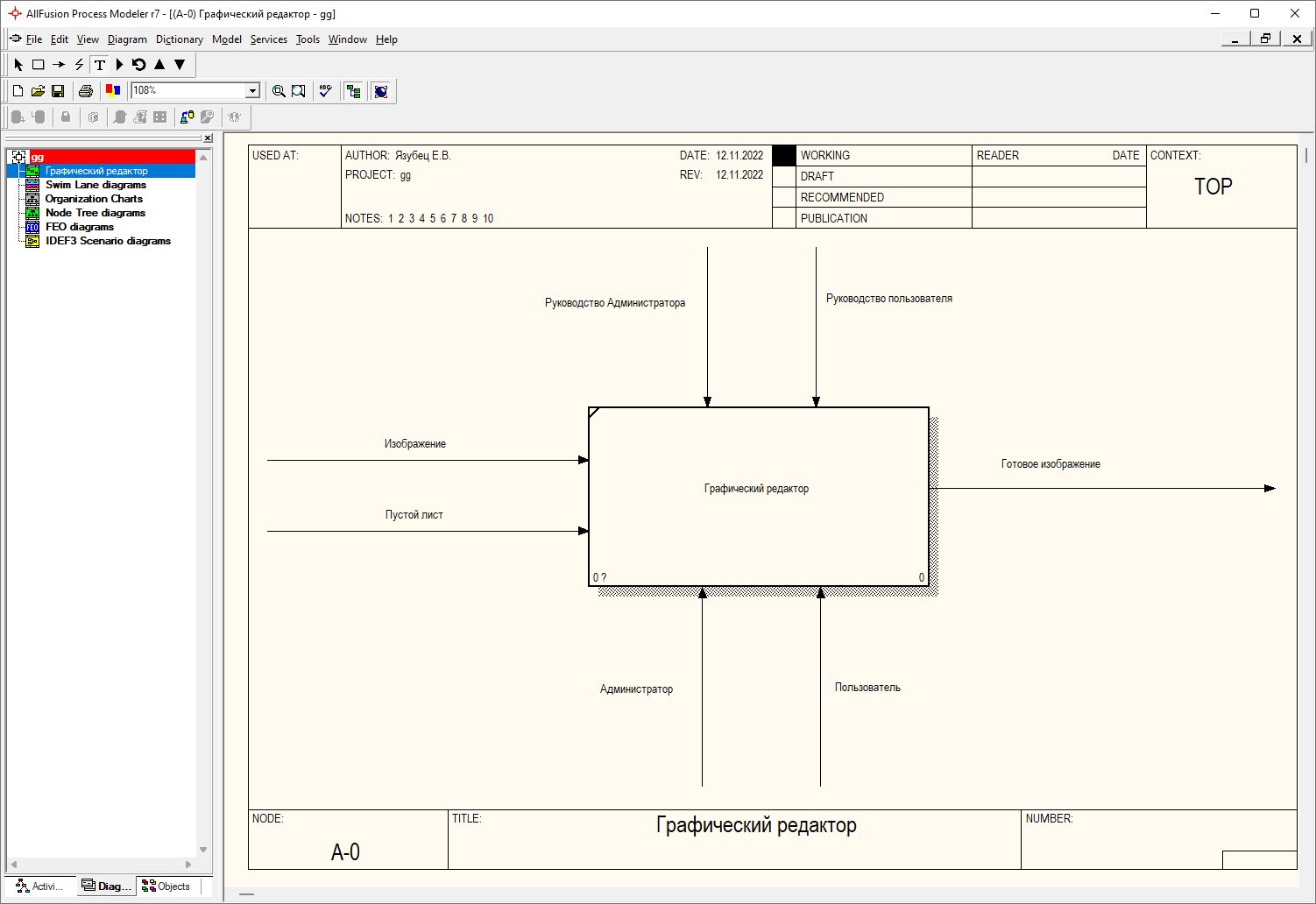
Появляется диалог Activity Box Count в котором выбираем тип диаграммы (декомпозицию можно провести в другой методологии)

и число активностей на диаграмме декомпозиции

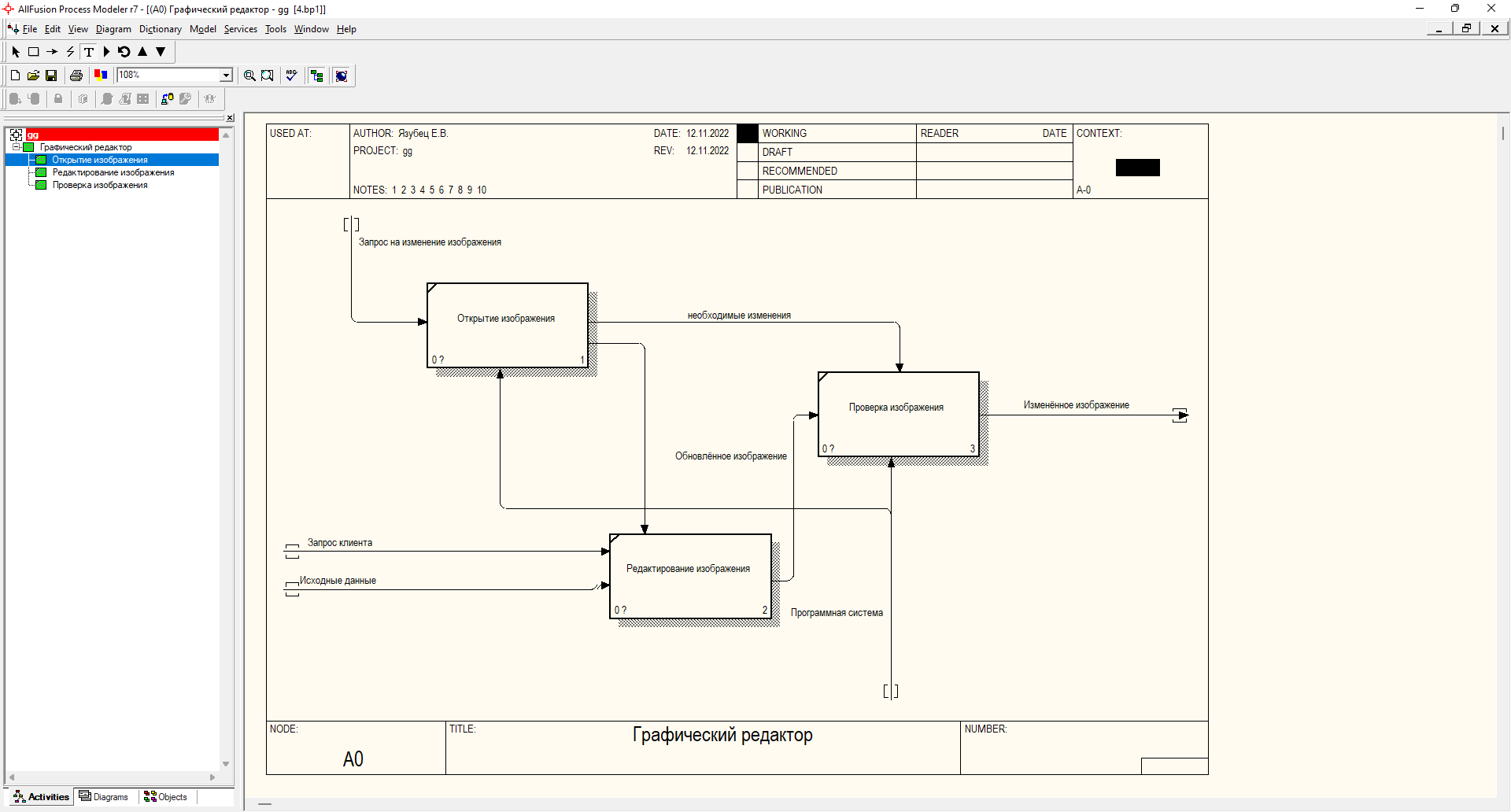
Выберем методологию IDEF0, зададим число активностей, равное 3 и нажмем кнопку OK. Появляется диаграмма декомпозиции с несвязанными граничными стрелками.

Далее создаёт диаграмму по нужной теме.

**2.Контекстная диаграмма**



**3.Декомпозиция функциональной диаграммы**



**4.Контрольные вопросы**

**1. Понятие Case-средств и их назначение.**

CASE средства (Computer - Aided Software Engineering) – это инструмент, который позволяет автоматизировать процесс разработки информационной системы и программного обеспечения.Основной целью применения CASE средств является сокращение времени и затрат на разработку информационных систем, и повышение их качества

**2. Назначение и сущность методологии IDEF0.**

**IDEF0** — **методология** функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. В **IDEF0** рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

**3. Назначение и сущность методологии DFD.**

**DFD** — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется **методология** графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

**4. Назначение и сущность методологии IDEF3.**

**DEF3** — способ описания процессов с использованием структу­рированного метода, позволяющего эксперту в предметной области представить положение вещей как упорядоченную последователь­ность событий с одновременным описанием объектов, имеющих не­посредственное отношение к процессу.

IDEF3 является технологией, хорошо приспособленной для сбора данных, требующихся для проведения структурного анализа системы.

**5. Направления IDEF0-моделирования.**

**DEF0** - методология функционального моделирования. С помощью наглядного графического языка IDEF0, изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков - в терминах IDEF0). Как правило, моделирование средствами IDEF0 является первым этапом изучения любой системы

**6. Этапы жизненного цикла программных средств, для которых**

**наиболее эффективно использование методологии IDEF0.**

Этапы жизненного цикла программных средств, для которых наиболее эффективно использование методологии IDEF0. Методология IDEF0 успешно применяется в самых различных отраслях как эффективное средство анализа, проектирования и представления деловых процессов.

**7. Достоинства методологии IDEF0.**

1. полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи);
2. комплексность при декомпозиции (мигрирование и туннелирование стрелок);
3. возможность агрегирования и детализации потоков данных и информации (разделение и слияние стрелок);
4. наличие жестких требований методологии, обеспечивающих получение моделей процессов стандартного вида;
5. простота документирования процессов; соответствие подхода к описанию процессов в IDEF0 стандартам ISO 9000:2000.

**8. Цель модели в IDEF0.**

Цель моделирования (Purpose). Модель не может быть построена без четко сформулированной цели. Цель должна отвечать на следующие вопросы:

Почему этот процесс должен быть замоделирован?

Что должна показывать модель?

Что может получить читатель?

Формулировка цели позволяет команде аналитиков сфокусировать усилия в нужном направлении. Примерами формулирования цели могут быть следующие утверждения: "Идентифицировать и определить текущие проблемы, сделать возможным анализ потенциальных улучшений", "Идентифицировать роли и ответственность служащих для написания должностных инструкций", "Описать функциональность предприятия с целью написания спецификаций информационной системы" и т. д.

**9. "Точка зрения" модели в IDEF0.**

Точка зрения (Viewpoint). Особенно важно включать в процесс разработки модели представителей различных мнений, однако сама модель должна базироваться на единой точке зрения. Чаще всего разнообразные точки зрения кратко фиксируют на диаграмме ФЕО (FEO – For Exposition Only – только для комментариев). Эти диаграммы используются только в качестве материалов для презентаций.

Точка зрения должна формулироваться исходя из цели построение диаграммы. При построении модели важно придерживаться одной точки зрения, которая должна содержать наименование должности, структурного подразделения или описание должностных обязанностей работника. Модели могут содержать разнообразные точки зрения с целью детальной фиксации всех действий (функций).

**10. Субъект моделирования в IDEF0. Принцип ограничения субъекта.**

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, мы должны определить, что мы будем в дальнейшем рассматривать как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будет существенно влиять позиция, с которой рассматривается система, и цель моделирования - вопросы, на которые построенная модель должна дать ответ, другими словами, первоначально необходимо определить область (Scope) моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели.

Хотя предполагается, что в течение моделирования область может корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования и когда должна быть закончена модель. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели - мы определяем, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком Уровне детализации модель является завершенной.

При определении глубины системы необходимо не забывать об ограничениях времени -трудоемкость построения модели растет в геометрической прогрессии от глубины декомпозиции. После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему; поскольку все объекты модели взаимосвязаны, внесение нового объекта может быть не просто арифметической добавкой, но в состоянии изменить существующие взаимосвязи. Внесение таких изменений в готовую модель является, как правило, очень трудоемким процессом (так называемая проблема "плавающей области").

**11. Правила представления работ на IDEF0-диаграмме.**

В основе методологии IDEF0 лежат следующие правила:

1) функциональный блок преобразует входы в выходы;

2) управление ограничивает или предписывает условия выполнения преобразований;

3) механизмы показывают, кто, что и как выполняет эти преобразования (т.е. механизмы непосредственно осуществляют эти преобразования).

**12. Назначения сторон функциональных блоков на IDEF0-диаграмме.**

Каждая из четырех сторон функционального блока имеет своё определенное значение (роль):

Верхняя сторона имеет значение “Управление” (Control);

Левая сторона имеет значение “Вход” (Input);

Правая сторона имеет значение “Выход” (Output);

Нижняя сторона имеет значение “Механизм” (Mechanism).

**13. Принцип доминирования и его представление на IDEF0-диаграмме.**

1. Функциональный блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных между собой дугами.

2. Эти блоки представляют основные подфункции (подмодули) единого исходного модуля.

3. Данная декомпозиция выявляет полный набор подмодулей, каждый из которых представлен как блок, границы которого определены дугами.

4. Каждый из этих подмодулей может быть декомпозирован подобным же образом для более детального представления.

**14. Назначение связей на IDEF0-диаграмме.**

В методологии IDEF0 используется пять типов взаимосвязей между блоками для описания их отношений: управление, вход, обратная связь по управлению, обратная связь по входу, выход-механизм.

**15. Описание связей на IDEF0-диаграмме.**

Самая верхняя диаграмма, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется A-0 (А минус нуль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой. Диаграмма A-0 устанавливает область моделирования и ее границу.

**16. Виды отношений между блоками и дугами на IDEF0-диаграмме.**

Разветвления дуг обозначают, что все содержимое дуг или его часть может появиться в каждом ответвлении дуги. При слиянии дуг результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения.

**17. Типы взаимосвязей между блоками на IDEF0-диаграмме.**

Отношение управления возникает тогда, когда выход одного блока непосредственно влияет на работу блока с меньшим доминированием.

Отношение входа возникает тогда, когда выход одного блока становится входом для блока с меньшим доминированием. Обратные связи по управлению и по входу представляют собой итерацию или рекурсию.

Обратная связь по управлению возникает тогда, когда выход некоторого блока влияет на работу блока с большим доминированием. Обратная связь по входу имеет место тогда, когда выход одного блока становится входом другого блока с большим доминированием. Связь "выход-механизм" встречается нечасто и отражает ситуацию, при которой выход одной функции становится средством достижения цели для другой функции. Данная связь характерна при распределении источников ресурсов (например, физическое пространство, оборудование, финансирование, материалы, инструменты, обученный персонал и т.п.).

**18. Разветвления дуг и правила их обозначения на IDEF0-диаграмме.**

Дуга всегда помечается до разветвления, чтобы дать название всему набору. Каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами:

– непомеченные ветви содержат все объекты, указанные в метке дуги перед разветвлением;

– каждая метка ветви указывает, что именно содержит ветвь.

**19. Слияние дуг и правила их обозначения на IDEF0-диаграмме.**

Каждая ветвь перед слиянием помечается или нет в соответствии со следующими правилами:

– непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния;

– метка ветви указывает, что конкретно содержит ветвь.

**20. Понятие диаграммы декомпозиции, родительского блока, родительской диаграммы в IDEF0-модели.**

Диаграмма декомпозиции предназначены для детализации функций и получаются при разбиении контекстной диаграммы на крупные подсистемы (функциональная декомпозиция) и описывающие каждый подсистему и их взаимодействие. Декомпозируемый блок называется родительским блоком, а содержащая его диаграмма – родительской диаграммой.

**21. Контекстная диаграмма модели.**

**Контекстная диаграмма** - вид IDEF0-**диаграммы**. Это **диаграмма**, расположенная на вершине древовидной структуры **диаграмм**, представляющая собой самое общее описание системы и ее взаимодействие с внешней средой (как правило, здесь описывается основное назначение моделируемого объекта).

**22. Номер узла IDEF0-диаграммы. Назначение и правила записи.**

Каждая диаграмма модели идентифицируется "номером узла" (NODE), расположенным на IDEF0-бланке в левом нижнем углу. Номер узла для контекстной диаграммы имеет следующий вид: заглавная буква A (Activity в функциональных диаграммах), дефис и ноль. Номер узла диаграммы, декомпозирующей контекстную диаграмму, – тот же номер узла, но без дефиса.

**23. Граничные дуги IDEF0-диаграммы и система их обозначений.**

Правила стыковки и обозначения внешних дуг диаграммы-потомка с граничными дугами родительского блока могут быть сформулированы следующим образом:

– зрительно соединяется каждая внешняя дуга диаграммы-потомка с соответствующей граничной дугой родительского блока;

– каждой зрительной связи присваивается код (I – для входных дуг, C – для связей между дугами управления, O – для связей между выходными дугами, M – для связей между дугами механизм;

-после каждой буквы добавляется цифра, соответствующая положению данной дуги среди других дуг того же типа, касающихся родительского блока. Входные и выходные дуги пересчитываются сверху вниз, а дуги управлений и механизмов пересчитываются слева направо (в том порядке, как они расположены на родительской диаграмме по отношению к родительскому блоку). Например, внешние дуги М1, М2 пронумерованы в соответствии с расположением граничных дуг на родительском блоке

**24. Тоннелирование связей. Назначение и правила обозначения.**

Особые ситуации возникают, когда дуги "входят в тоннель" между диаграммами.

Дуга "входит в тоннель", если:

1) она является внешней дугой, которая отсутствует на родительской диаграмме (дуга имеет скрытый источник);

2) она касается блока, но не появляется на диаграмме, которая его декомпозирует (имеет скрытый приемник).

Тоннельные дуги от скрытого источника начинаются скобками, чтобы указать, что эти дуги идут из какой-то другой части модели, прямо извне модели или они не важны для родительской диаграммы и поэтому на ней не изображаются. Например, дуга механизма «Лаборант» является тоннельной дугой со скрытым источником. Ее нет на родительской диаграм, поскольку для родительской диаграммы данная дуга является маловажной.

Тоннельные дуги со скрытым приемником заканчиваются скобками, чтобы отразить тот факт, что такая дуга идет к какой-то другой части модели, выходит из нее или не будет более в этой модели рассматриваться. Тоннельные дуги со скрытым приемником часто используются в том случае, если данные дуги должны связываться с каждым блоком диаграммы-потомка. Изображение таких дуг может существенно загромоздить данную диаграмму. Например, дуги «Предварительные знания» и «Порядок выполнения работы» должны связываться с каждым блоком диаграммы декомпозиции. Их изображение на диаграмме-потомке является малоинформативным. Поэтому данные дуги реализованы как тоннельные дуги со скрытым приемником и на диаграмме декомпозиции не показаны.

Таким образом, "Вхождение дуг в тоннель" используется, как правило, для упрощения описания системы – тогда, когда диаграммы в модели становятся слишком трудными для чтения и понимания.

**25. Основные этапы процесса моделирования в IDEF0.**

Процесс моделирования в IDEF0 включает сбор информации об исследуемой области, документирование полученной информации с представлением ее в виде модели и уточнение модели посредством итеративного рецензирования.

**26. Методологии, поддерживаемые BPwin.**

BPwin поддерживает три методологии — IDEF0, IDEF3 и DFD.

**27. Последовательность действий по созданию IDEF0-модели в**

**BPwin.**

Для внесения субъекта, цели и точки зрения модели IDEF0 в BPwin необходимо выбрать пункт меню Edit / Model Properties (Свойства модели), вызывающий диалог Model Properties. В закладке Purpose следует

указать цель и точку зрения. Закладка Purpose диалога Model Properties В закладке Definition необходимо определить субъект моделирования (Definition) и его границы (Scope). В закладке Status определяется статус

модели (черновой, рабочий, окончательный и т.д.), время создания или последнего редактирования. В закладке Source описываются источники информации для построения модели. Закладка General служит для внесения имени проекта и модели, фамилии и инициалов автора и вида модели - AS-IS или TO-BE

**28. Назначение пунктов главного меню BPwin.**

Главное меню содержит пункты File, ModelMart, Edit, Tasks, Client, Server, Option, Window, Help.

В состав меню Файл входят следующие команды: •

New - создает новый Erwin-файл данных; •

Open - позволяет открыть уже существующий Erwin-файл; •

Close - закрывает текущий файл; • Save - сохраняет текущий файл;

Save as - сохраняет текущий файл под другим именем;

Dictionary manager - позволяет выбрать язык, на котором будет создан SQL скрипт;

Print и Print Setup - осуществляют настройку печати;

BPWin и Designer/2000 - осуществляют взаимодействие с CASEсредствами BPWin и Designer/2000.

Exit - завершает сеанс работы с Erwin;

Команды меню ModelMart обеспечивают согласованное проектирование БД и приложений при коллективной разработке в сетевой версии Erwin.

Команды меню Edit позволяют:

редактировать объекты схемы (под объектами в данном случае понимаются сущности, атрибуты, связи и др. в логической модели или таблицы, колонки, связи, индексы, триггеры и др. в физической модели);

* создавать новые объекты;
* перерисовывать диаграмму;
* копировать объекты;
* вставлять объекты;
* переходить к заданному объекту
* Команды менюTasks предназначены для выполнения следующих задач:
* генерация базы данных;
* обратное (реинженеринг) проектирование;
* синхронизация с базой данных;
* генерация отчетов Client входят следующие команды:

PB Edit Styles - позволяет создать новый или изменить существующий стиль редактирования данных. Этот стиль редактирования используется PowerBuilder для представления данных пользователю. Стиль также определяет, каким способом пользователь может редактировать или выбирать данные;

PB Display Format - позволяет создать новый или изменить существующий формат представления данных. Формат данных представляет собой фильтр, который позволяет вводить данные только по заданной маске. Маска задается с помощью цифр и букв, где цифры и буквы имеют специальное значение;

Validation Rule - позволяет создать новое или изменить существующее правило, по которому будет происходить проверка введенных данных. Это правило накладывает определенные ограничения на тип и значение вводимых пользователем данных. В появившемся диалоговом окне можно ввести сообщение, которое будет появляться, когда пользователь базы данных ввел неправильные значения. Можно установить опции, которые определяют, проверяются данные на сервере или в клиентском приложении;

Valid value - позволяет создать новые или изменить существующие области значений, в соответствии с которыми будут проверяться данные, вводимые пользователем в поля баз данных;

Default/Initial - позволяет создавать новые или изменять существующие способы заполнения полей базы данных по умолчанию. При добавлении в базу новой записи поля этой записи по умолчанию могут принять одно из созданных значений;

Target client - позволяет выбрать то средство разработки 4GL, которое будет использовать сгенерированную Erwin-схему (в данном случае выбран программный продукт PowerBuilder);

PB Sync option - позволяет настроить опции, которые определяют, какие данные и как синхронизировать с PowerBuilder;

Sync ERWin with PB - выполняет синхронизацию ERWin с PowerBuilder. Для того, чтобы синхронизация прошла успешно, необходимо наличие драйверов ODBC.

Описание команд меню Server

СУБД Schema properties позволяет определить SQL скрипт, который будет выполняться до соединения и после соединения с базой данных на сервере;

СУБД Trigger Template позволяет задать вид действия, которое будет выполнено по отношению к главной или подчиненной таблице при определенных действиях пользователя с данными таблицы. Например, 71 при удалении записи из главной таблицы запись из подчиненной таблицы тоже будет удалена;

Validation Rule, Valid value, Default/Initial выполняют действия,

аналогичные действию соответствующих команд меню Client, но для серверной части;

Target Server позволяет выбрать необходимую СУБД с помощью диалогa;

СУБД Connection позволяет подсоединиться к серверу СУБД. Для соединения необходимо наличие драйверов. Меню Option предоставляет команды, с помощью которых пользователь может по своему желанию настроить опции, влияющие на внешний вид окна ERWin и ER-диаграммы.

**29. Установка размеров полей стандартного бланка диаграммы в BPwin.**

Пункт меню Page Setup предназначен для установки размеров полей стандартного бланка диаграммы. Данный пункт содержит подпункты установки размеров полей для текущей диаграммы, для новой диаграммы и для новой модели.

**30. Правила внесения субъекта, его границ, цели и точки зрения моделиIDEF0 в BPwin.**

Для внесения субъекта, цели и точки зрения модели IDEF0 в BPwin необходимо выбрать пункт меню Edit / Model Properties (Свойства модели), вызывающий диалог Model Properties. В закладке Purpose следует указать цель и точку зрения.

**31. Назначение видов модели AS-IS или TO-BE в BPwin.**

Модель AS-IS позволяет определить неэффективные места существующего на момент моделирования процесса, оценить, насколько глубоким изменениям необходимо подвергнуть существующую структуру организации системы. Признаками неэффективности существующего процесса могут быть, например, бесполезные работы (в работах отсутствует выход), неуправляемые работы (в работах отсутствует управление) и дублирующийся работы, отсутствие обратных связей по управлению (на проведение процесса не оказывает влияния его результат), входу (материалы или информация используются нерационально).

С учетом анализа найденных в модели AS-IS недостатков создаются модели ТО-ВЕ. Модели ТО-ВЕ используются для оценки более эффективных способов выполнения процесса в системе. На основе модели ТО-ВЕ, отражающей оптимальный способ выполнения процесса, строится прототип, а затем окончательный вариант системы.

**32. Правила создания контекстной диаграммы модели в BPwin.**

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции.

После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступать к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом и каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

**33. Назначение кнопок палитры инструментов для IDEF0-методологии вBPwin.**

Вид кнопки Назначение кнопки

Добавить в диаграмму внешнюю ссылку

Ссылка на другую страницу (off-page reference), позволяет направить стрелку на любую диаграмму

Добавить в диаграмму хранилище данных (Data store)

**34. Правила ус Вид кнопки Назначение кнопки**

Чтобы не возникло проблем с переходом к русской раскладке клавиатуры, следует в рабочей области работы предварительно щелкнуть правой кнопкой мыши, в появившемся контекстном меню выбрать пункт Font Editor и затем шрифт Times New Roman

**35. Правила создания граничных связей в BPwin.**

Для создания граничной входной дуги необходимо:

щелкнуть по кнопке с символом стрелки (режим рисования стрелок) в палитре инструментов, перенести курсор к левой стороне экрана до появления левой границы диаграммы, выделенной полосой;

щелкнуть один раз по левой границе диаграммы (отмечается место, откуда выходит стрелка), затем по левой границе работы (отмечается место, где заканчивается стрелка);

вернуться в палитру инструментов и для присваивания стрелке названия выбрать в палитре инструментов режим редактирования;

щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню редактирования связей выбрать пункт Name Editor и в появившемся диалоге IDEF0 Arrow Properties внести имя дуги и фамилию автора диаграммы; основу названия дуги на IDEF0- диаграммах должно составлять существительное (например, «Код программы», «Студент», «Результаты», «График выполнения»).

**36. Правила создания диаграмм декомпозиции в BPwin.**

Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую. Для создания диаграммы декомпозиции необходимо левой кнопкой мыши выделить родительскую работу и щёлкнуть по кнопке «Декомпозиция» палитры инструментов. В результате возникает диалог Activity Box Count, в котором следует указать нотацию новой диаграммы и количество работ на ней. В итоге будет получена диаграмма декомпозиции, содержащая пять работ и несвязные стрелки. Несвязными стрелками являются дуги, касающиеся декомпозированного блока родительской диаграммы.

Для связывания граничных стрелок выхода с соответствующими работами необходимо в режиме редактирования стрелок щелкнуть по соответствующей стороне работы и затем по наконечнику стрелки. Для связи работ между собой используются внутренние стрелки, т.е. стрелки, начинающиеся у одной и кончающиеся у другой работы и не касающиеся границ диаграммы. Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по стороне выхода работы источника стрелки и затем по соответствующей стороне (входа, управления или 31 механизмов) работы-приемника стрелки.

Для разветвления стрелки следует в режиме рисования стрелок щелкнуть по сегменту стрелки, которую нужно разветвить, и затем по соответствующей стороне (входа, управления или механизмов) работы-приемника ветви стрелки. Для слияния стрелок следует в режиме рисования стрелок щелкнуть по стороне выхода работы-источника ветви стрелки и затем по сегменту стрелки, которую нужно слить с ветвью. Для удаления блока (стрелки) необходимо его (ее) выделить с помощью мыши и нажать на клавишу Del. Для идентификации граничных стрелок диаграммы используются коды ICOM. Для отображения кодов ICOM на диаграммах модели необходимо включить опцию Show ICOM Codes на закладке Presentation диалогового окна Molel Properties.

Данное окно может быть вызвано с помощью меню Edit/Model Properties или нажатием правой кнопки мыши в свободной области диаграммы. Если в результате разработки диаграммы декомпозиции окажется, что в нее необходимо добавить работу, то для этого следует войти в режим рисования работ и щелкнуть в нужном месте рабочего поля диаграммы левой кнопкой мыши. В результате в выбранном месте диаграммы появится новая работа. Для присвоения работам и стрелкам имен необходимо выполнить действия, аналогичные описанным для контекстной диаграммы.

**37. Правила создания, разветвления и слияния граничных связей в BPwin.**

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

необходимости для некоторых диаграмм декомпозиции строятся диаграммы FEO, отображающие альтернативную точку зрения.

2. При разработке моделей следует избегать

изначальной «привязки» функций исследуемой системы к существующей организационной структуре моделируемого объекта (предприятия, фирмы). Это помогает избежать субъективной точки зрения, навязанной организацией и ее руководством. Организационная структура должна явиться результатом использования (применения) модели.

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

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

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

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

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

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

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

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

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

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

13. Если две стрелки проходят параллельно (начинаются из одной и той же грани одной функции и заканчиваются на одной и той же грани другой функции), то по возможности следует их объединить и дать единое обозначение.

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

**38. Правила тоннелирования связей в BPwin.**

В некоторых случаях удобно использовать механизм тоннелирования связей. Существует два вида тоннелирования связей: со скрытым приемником и со скрытым источником. Связь со скрытым приемником удобно использовать, если связь, входящую в родительский блок, нежелательно изображать на диаграмме декомпозиции. Например, если граничная связь управления или механизмов поступает на все работы диаграммы, она может быть

**39. Правила рисования диаграмм в BPwin.**

Прямоугольники работ должны располагаться по диагонали с левого верхнего в правый нижний угол (порядок доминирования). При создании новой диаграммы декомпозиции BPwin автоматически располагает работы именно в таком порядке. В дальнейшем можно добавить новые работы или изменить расположение существующих, но нарушать диагональное расположение работ по возможности не следует.

Порядок доминирования подчеркивает взаимосвязь работ, позволяет минимизировать изгибы и пересечения стрелок. Следует максимально увеличивать расстояние между входящими или выходящими стрелками на одной грани работы. Если включить опцию Line Drawing: Automatically space arrows на закладке Layout диалога Model Properties (меню Edit/Model Properties), BPwin будет располагать стрелки нужным образом автоматически. Следует максимально увеличить расстояние между работами, поворотами и пересечениями стрелок.

Если две стрелки проходят параллельно (начинаются из одной и той же грани одной работы и заканчиваются на одной и той же грани другой работы), то по возможности следует их объединить и назвать единым термином. Обратные связи по входу рисуются "нижней" петлей, обратная связь по управлению - "верхней". BPwin автоматически рисует обратные связи нужным образом. Его можно "обмануть", но лучше этого не делать. Циклические обратные связи следует рисовать только в случае крайней необходимости, когда подчеркивают значение повторно используемого объекта. Принято изображать такие связи на диаграмме декомпозиции. BPwip не позволяет создать циклическую обратную связь за один прием. Если все же необходимо изобразить такую связь, следует сначала создать обычную связь по входу, затем разветвить стрелку, направить новую, ветвь обратно ко входу работы-источника и, наконец, удалить старую ветвь стрелки выхода.

**40. Диаграммы дерева узлов и правила их создания в BPwin.**

Основная идея методологии SADT - построение древовидной функциональной модели предприятия. Сначала функциональность предприятия описывается в целом, без подробностей. Такое описание называется контекстной диаграммой. Взаимодействие с окружающим миром описывается в терминах входа (данные или объекты, потребляемые или изменяемые функцией), выхода (основной результат деятельности функции, конечный продукт), управления (стратегии и процедуры, которыми руководствуется функция) и механизмов (необходимые ресурсы).

Кроме того, при создании контекстной диаграммы формулируются цель моделирования, область (описание того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и точка зрения (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственного за работу моделируемой системы в целом. Затем общая функция разбивается на крупные подфункции.

Этот процесс называется функциональной декомпозицией. Затем каждая подфункция декомпозируется на более мелкие - и так далее до достижения необходимой детализации описания. На рис. 1 показано дерево функций, называемое деревом узлов функциональной модели. Каждый узел соответствует отдельному фрагменту описания - диаграмме. Модель представляет собой совокупность иерархически выстроенных диаграмм, каждая из которых является описанием какой-либо функции или работы (activity).