Практическая работа №2. Анализ бизнес-процессов в методологии SADT

Цель работы: получить практические навыки анализа бизнес-процессов в методологии SADT в нотациях IDEF0 и DFD.

Задание

Использовать предприятие практической работы 1.

1. Разработать модель процессов в нотации IDEF0.
2. Создать словарь данных модели
3. Дополнить модель DFD диаграммами

Указания

В работе используется свободно распространяемое ПО Ramus2.0 https://github.com/Vitaliy-Yakovchuk/ramus/releases или Ramus Education

1. Разработка модели процессов в нотации IDEF0.

Методология IDEF0 (Integrated DEFinition) представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели предметной области. Данная методология отображает функциональную структуру системы, т.е. производимые ею действия и связи между этими действиями.

Данная методология применяется при создании новых систем для определения требований и функций и затем для разработки системы, удовлетворяющей требованиям и реализующей функции. Для действующих систем эта методология может использоваться для анализа функций, выполняемых системой, а также для наглядного представления «механизмов», посредством которых эти функции осуществляются. Основной сферой применения методологии IDEF0 является предпроектное обследование и анализ системы.

Методология IDEF0 основана на методе SADT Дугласа Т. Росса. В рамках проекта ICAM, организованного военными ведомствами США с целью разработки подходов, обеспечивающих повышение эффективности проектирования благодаря систематическому внедрению компьютерных технологий, метод SADT и некоторые аспекты его применения были стандартизированы, после чего получили название методологии IDEF0.

Модель IDEF0 представляет собой набор диаграмм с поддерживающей их документацией, включающей сопровождающие тексты и словарь. Диаграммы модели декомпозируют сложную систему на составные части. Первоначальная (исходная, корневая) диаграмма является наиболее общим и наиболее абстрактным описанием всей системы в целом. Она показывает основную функциональную составляющую системы в виде блока. Детали (компоненты) каждого из основных блоков показаны на других диаграммах в виде блоков. Далее они могут быть превращены в более подробные

диаграммы, до тех пор, пока не будет достигнута требуемая степень детализации.

Блоки представляют собой функции (действия, процессы или операции), а входящие и исходящие из них стрелки представляют объекты (информацию, предметы). Блок и его стрелки на диаграмме рассматриваемого уровня описывается более подробно блокам и стрелками диаграмм нижнего уровня.

Блоки-функции показывают, что должно выполняться, причем без идентификации каких-либо других аспектов, например, таких как потребности в них или средства их осуществления. Наименование функций записывается внутри блока. Оно должно содержать существительное, обозначающее действие. Блоки нумеруются в нижнем углу.

Стрелки на диаграмме играют роль интерфейсов (связей) блоков с внешней для них средой. Каждая из стрелок имеет метку, характеризующую её. Назначение стрелок зависит от стороны блока, в которую стрелка входит или выходит (рис. 1):

Входящие стрелки слева от блока представляет собой предметы (материальные объекты) или информацию (информационные объекты), необходимые для выполнения функции. Это сырье, материалы, исходные данные или «вход» функции (стрелка типа I – input).

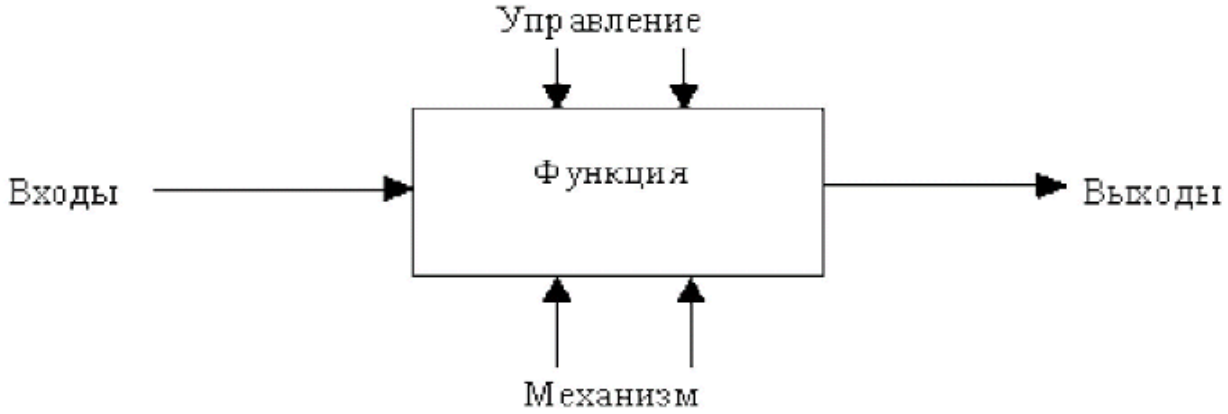

Рисунок 1 – Функциональный блок и интерфейсные дуги

Выходящие стрелки справа из блока показывают предметы или данные, полученные в результате выполнения функции блока. Это результат (стрелки типа O – output). Функции преобразуют объекты слева направо (от входа к выходу). Таким образом, блок представляет собой переход от состояния «до» к состоянию «после».

Входящие стрелки сверху блока является управлением, описывающим условия или данные, которые управляют выполнением функции (тип C – control). Как и входы, они тоже могут являться информацией, но назначение входа и управления различны. Их разделение является важным для понимания работы системы.

Входящая снизу стрелка представляет собой «механизм», обозначающая собой либо человека, либо некоторое средство, выполняющее функцию (M – mechanism).

Выход и вход показывают, что и из чего делается функцией, управление показывает, как и почему это делается, а механизм показывает, кем и с помощью чего это делается.

Стрелки на диаграмме IDEF0 означают ограничения, задаваемые связанными с ними объектами (предметами или данными). Они не представляют собой поток или последовательность. Соединяя выход одного блока с входом другого, они показывают ограничения. Блок, получающий объекты, «ограничен» в том смысле, что функция не может быть выполнена, пока не будут получены объекты, производимые другими блоками. Стрелки, входящие в блок, показывают все объекты, которые необходимы для выполнения функции.

Несколько функций на диаграмме могут выполняться одновременно, если удовлетворены все ограничивающие условия. Ни последовательность, ни время не являются точно определенными в IDEF0. Обратная связь, итерация, непрерывные процессы, перекрытие (во времени) функций легко отображаются стрелками.

Стрелки могут разветвляться или соединяться. Каждая из ветвей может представлять один и тот же объект или различные объекты одного и того же типа.

Отсюда следует, что IDEF0-модели – это ни блок-схемы, ни просто диаграммы потоков данных, а предписывающие диаграммы, которые представляют входные/выходные преобразования, а также указывают правила этих преобразований. Примеры типичных наименований блоков и стрелок приведены в следующей таблице.

Функции		
планирование	разработка	классификация
управление	проектирование	измерение
контроль	изготовление	редактирование
расчет	производство	учет
продажа	реализация	сбор
Управление		
инструкции	требования	чертеж
стандарты	запросы	правило
указания	заявки	
задания	план	
Механизм		
персонал	компьютер	информационная система
фирма	станок	программное обеспечение
заказчик	аппаратура	подрядчик
инструмент	оператор	оснастка

Методологии IDEF послужили основой для создания нескольких CASE-продуктов. Один из них Ramus. Пакет Ramus – графическая среда для проектирования и моделирования сложных систем широкого назначения. Ramus Educational – это бесплатный аналог Ramus с некоторыми ограничениями (www.ramussoftware.com), который может быть использован для создания диаграмм в формате IDEF0 и DFD. Формат файлов Ramus Education полностью совместим с форматом файла коммерческой версии Ramus.

Общая схема составления функциональной модели состоит из следующих этапов:

1 построение модели (разработка функциональной диаграммы; заполнение глоссария дополнительными определениями; дополнение диаграммы гипертекстом);

2 проверка синтаксиса модели (проверка на наличие связей, на идентификаторы функций и связей, на управление).

Чтобы начать создание новой IDEF0-модели необходимо:

1 запустить Ramus, при первом запуске программа предлагает создать новый файл или открыть уже существующий (рис. 2);

2 после того как выбрано «Создать новый файл» в появившемся диалоговом окне «Мастер «Свойства проекта» для создания информационной модели нужно выбрать IDEF0 (рис 3);

3 подтвердить выбор клавишей «Окончить»;

4 откроется начальная IDEF0-страница (рис. 4) – контекстная диаграмма. В каждой IDEF0-модели можно иметь только одну контекстную диаграмму.

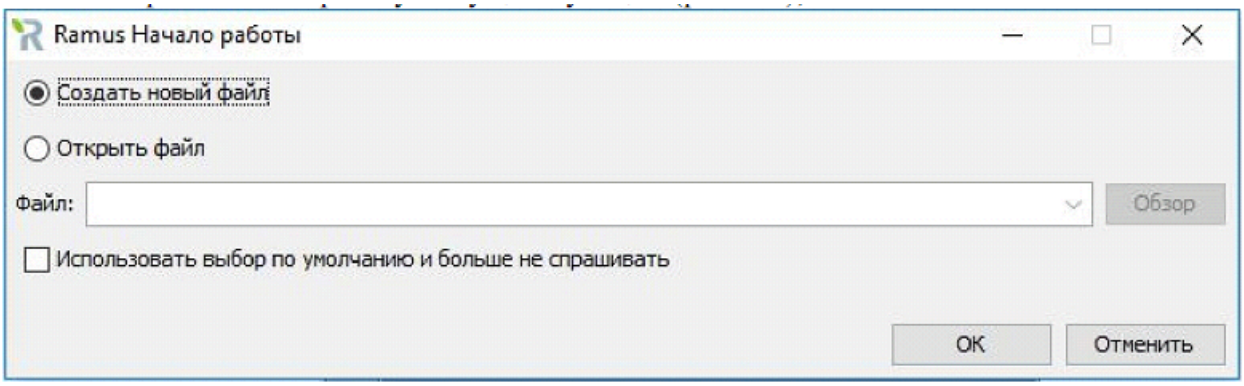

Рисунок 2 – Начало работы

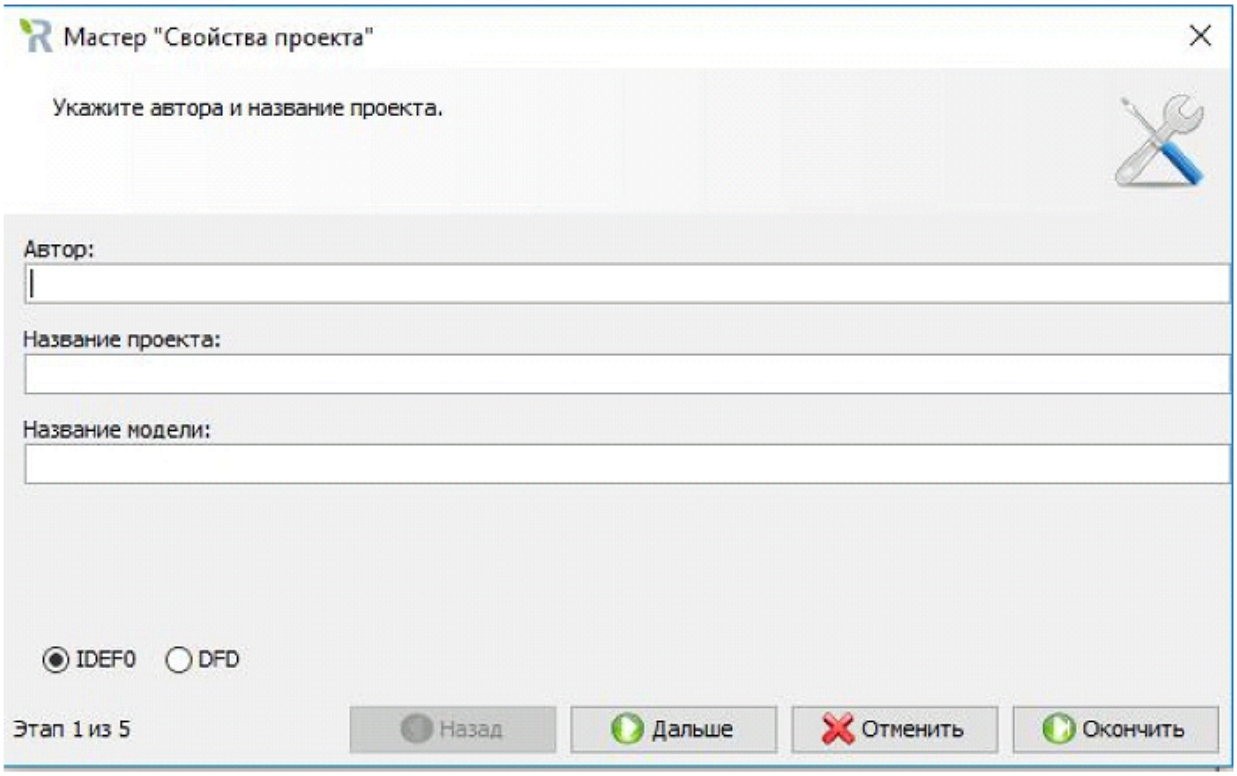

Рисунок 3 – Диалоговое окно «Мастер «Свойства проекта»

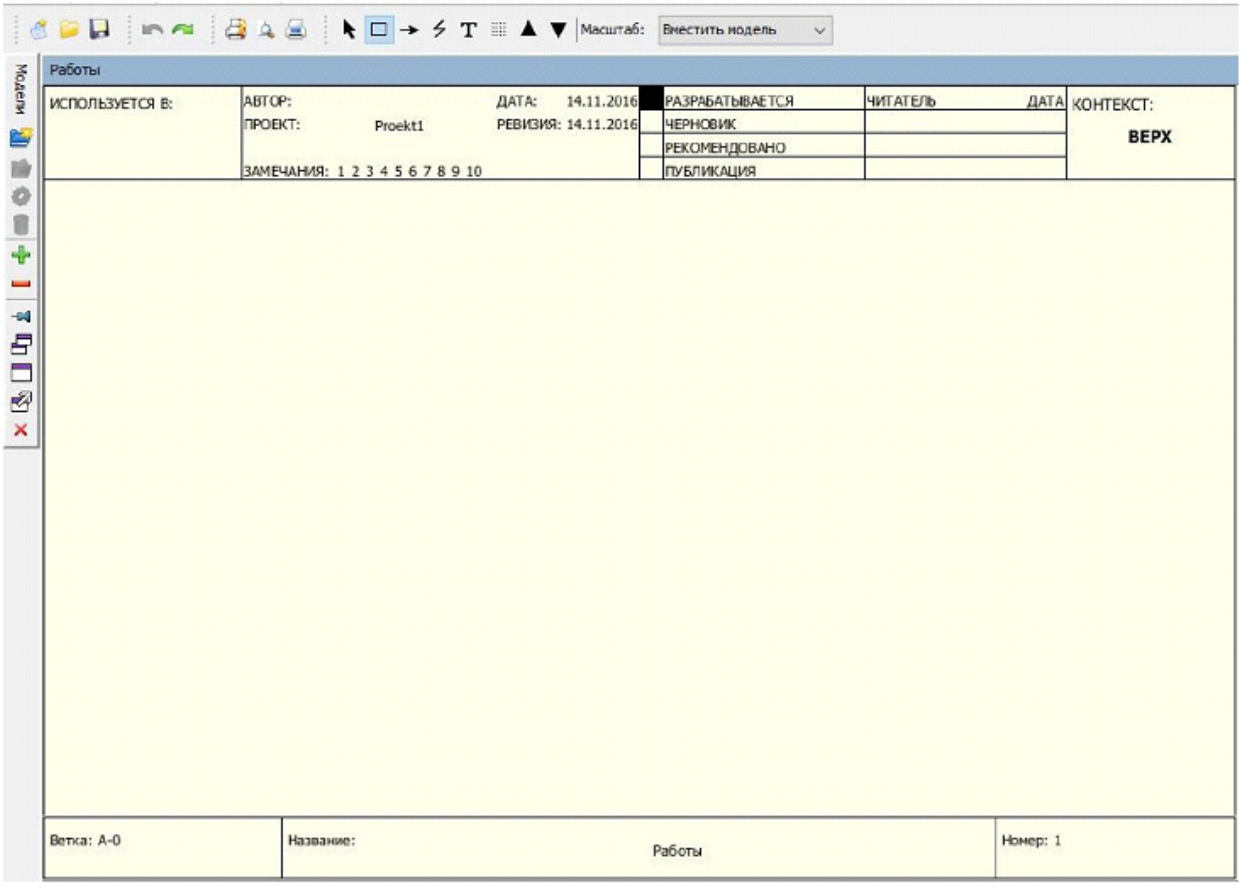

Рисунок 4 – Главное окно Ramus

Прежде, чем создавать остальные блоки, желательно сразу настроить среду проектирования, так как в противном случае в дальнейшем придется устанавливать настройки для каждого созданного блока в отдельности. Для этого в меню «Сервис», пункте «Свойства программы» на вкладке «Диаграмма» необходимо настроить все шрифты, которые будут действовать по умолчанию (рис. 5).

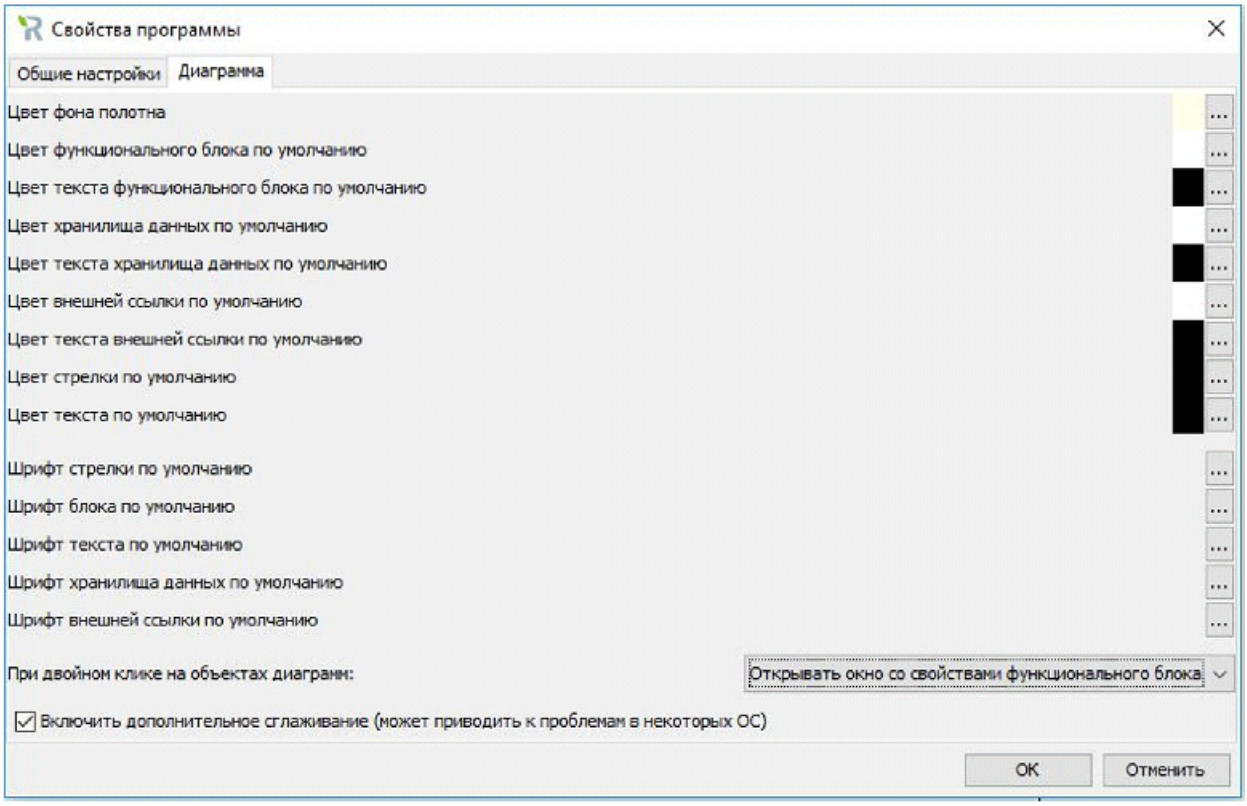

Рисунок 5 – Диалоговое окно «Свойства программы»

Для печати настройки устанавливаются в меню «Файл», пункте «Параметры страницы». Следует установить альбомное расположение листа, а также соответствующий формат. В настройках атрибутов проекта (Сервис□Свойства проекта□Автоматическое добавление атрибутов) устанавливаются соответствующие атрибуты.

Для добавления блока необходимо перейти в нужный режим путем вызова пункта «Режим добавления функциональных блоков» в меню «Диаграмма» или нажатием соответствующей кнопки на панели инструментов . После выбора пункта или нажатия кнопки следует указать место, где будет расположен новый блок. Начальная нумерация для блока контекстной диаграммы – «A0».

Размеры новых блоков устанавливаются по умолчанию, но их можно изменить путем растягивания за его границы. Каждый блок в диаграмме должен быть подписан. Правильным является именование блоков с помощью глаголов неопределенной формы. До начала ввода подписи можно заранее установить нужные параметры шрифта в меню «Сервис», пункте «Свойства программы» на вкладке «Диаграмма», либо изменить параметры во вкладке

«Шрифт» диалогового окна «Свойства функционального блока», которое вызывается двойным щелчком по самому блоку. Для того чтобы назвать блок, требуется выполнить одно из указанных действий:

 щелкнуть по функциональному блоку два раза и дать имя блоку во вкладке «Название»;

 щелкнуть правой кнопкой мыши по функциональному блоку и выбрать команду «Переименовать».

К функциональному блоку с четырех сторон должны быть присоединены различные объекты. Для ввода названий (ярлыков) объектов используется пункт «Режим добавления текста» меню «Диаграмма», клавиша B клавиатуры или кнопка T на панели инструментов. После нажатия следует указать место вставки ярлыка и ввести нужный текст так же, как и для функционального блока. Отмена режима ввода надписей происходит при нажатии на клавишу Esc или на любую кнопку на панели инструментов. Редактировать текст можно двойным кликом по нему, в появившемся диалогом окне «Параметры текста» можно изменить его шрифт и цвет.

Стрелки на функциональной диаграмме предназначены для соединения ярлыков с функциональными блоками и блоков между собой. Чтобы появилась возможность нарисовать стрелку можно воспользоваться одним из предложенных вариантов:

 выбрать команду «Режим работы со стрелками» пункта меню «Диаграмма»;

 нажать на кнопку на панели инструментов.

Далее нужно поместить курсор на границах модели до появления черной полосы (если стрелка идет извне в блок), либо у правой границы внутри функционального блока до появления черной стрелки (если стрелка идет из блока вовне). После чего протянуть курсор до появления черной стрелки внутри функционального блока, либо черной полосы у правой границы модели соответственно.

Стрелки на диаграмме могут сливаться в одну или наоборот разъединяться. Если несколько блоков должны быть соединены стрелками от одного источника, необходимо:

1 нарисовать стрелку к одному из блоков;

2 выделить ее в режиме работы со стрелками;

3 провести стрелку к нужному блоку.

Если несколько дуг от разных блоков должны соединяться в одну, необходимо:

1 нарисовать стрелку к одному из блоков;

2 выделить ее в режиме работы со стрелками;

3 провести следующую стрелку от блока к первой нарисованной стрелке (при наведении курсора на нее, она выделяется черным), две стрелки соединятся в одну.

Иногда на диаграмме требуется подписать стрелку, соединяющую функциональные блоки. Эта процедура выполняется в два этапа:

1 задать название стрелки, дважды щелкнув на нее и введя название в открывшемся диалоговом окне «Параметры стрелки» во вкладке «Поток»;

2 активизировать «Режим размещения тильд» в меню «Диаграмма», нажатием кнопки V на клавиатуре или // на панели инструментов, нажать на название стрелки и подвести курсор мыши к стрелке, которую требуется подписать. При этом стрелка выделится черным, после чего нажать на нее. По окончании процедуры от подписи к стрелке будет проведена волнистая линия, символизирующая их взаимную связь.

Для создания следующего функционального уровня модели необходимо пометить щелчком мыши функциональный блок и выполнить команду «Перейти к дочерним диаграммам» пункта меню «Диаграмма», либо выбрать кнопку на панели инструментов. . Появится диалоговое окно, в котором следует выбрать шаблон уровня и количество функциональных блоков (рис. 6).

На данной странице модели уже находятся ярлыки, созданные на предыдущем уровне. Для перехода на родительскую или дочернюю диаграммы используются пункты «Перейти к родительской диаграмме» и «Перейти к дочерним диаграммам» меню «Диаграмма» или кнопки и на панели инструментов.

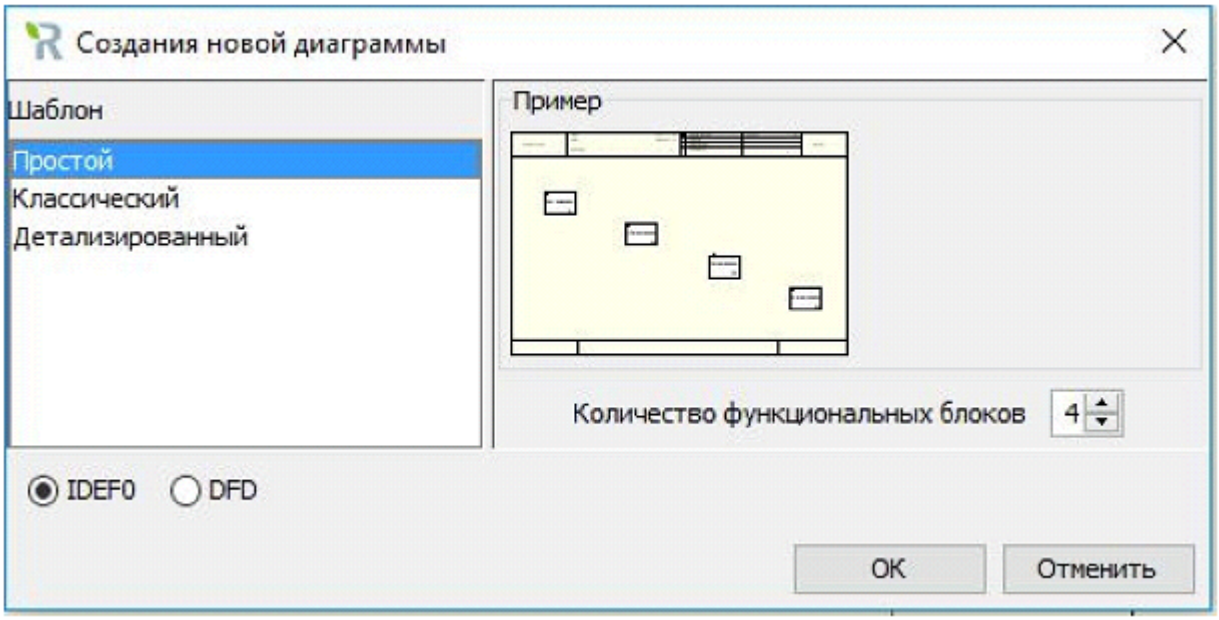

Рисунок 6 – Диалоговое окно «Создания новой диаграммы»

В программе предусмотрена возможность упорядоченного хранения информации о таких объектах, как персонал, документы, функции и т.д. в виде системы классификаторов. Классификация объектов значительно упрощает поиск и обработку информации об объектах модели. Каждый элемент системы классификации, кроме названия, может иметь дополнительные атрибуты, в которых можно упорядочено хранить разнообразнейшую информацию об объекте.

Заполнение классификатора производится во вкладке «Классификаторы» в правом верхнем углу модели либо нажатием «Окна» –

«Рабочее пространство» – «Классификаторы». Добавление нового классификатора происходит путем нажатия кнопки в окне «Классификаторы» в верхнем левом углу (рис. 7). Каждый классификатор должен содержать элементы, которые добавляются в центральном окне рабочего пространства.

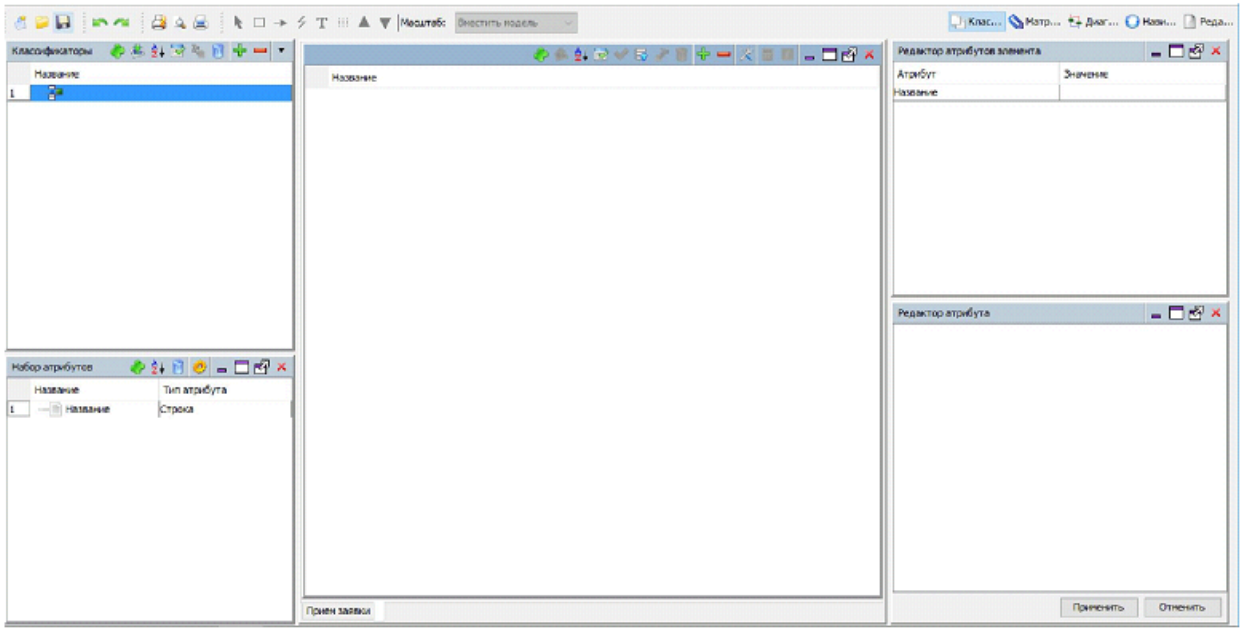

Рисунок 7 – Рабочее пространство «Классификаторы»

Элементы классификатора можно сортировать по названию, объединять и перемещать в другой классификатор с помощью соответствующий кнопок на панели инструментов. Также можно группировать элементы внутри классификатора, для этого необходимо нажать на номер элемента и, не отпуская курсор, перенести его на другой элемент. Появится папка, содержащая перенесенный элемент классификатора.

Классификаторы могут иметь разный набор атрибутов. Новый атрибут добавляется нажатием кнопки на панели инструментов окна «Набор атрибутов» в левом нижнем углу рабочего пространства. Появится окно «Добавление атрибута» (рис. 8), в нем необходимо выбрать тип атрибута и классификаторы, к которым он будет применяться.

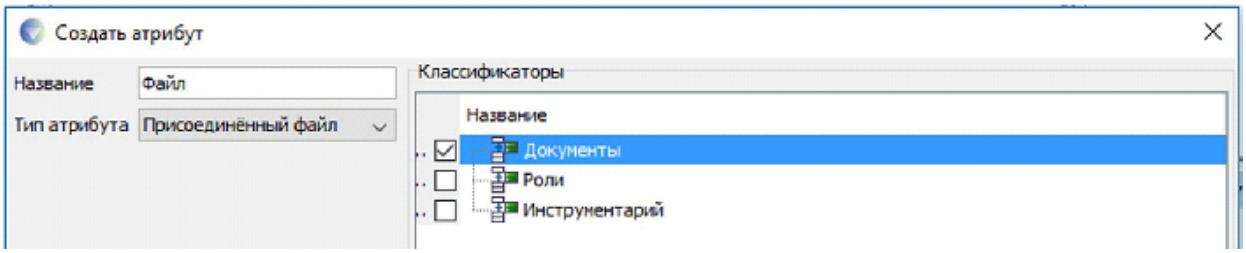

Рисунок 8 – Диалоговое окно «Создать атрибут»

Таким образом, появляется возможность дополнять элемент различной информацией (рис.9). Рассмотрим, какие еще типы атрибутов можно использовать. Для этого надо отредактировать атрибут с помощью

соответствующей кнопки на панели инструментов. Для удаления атрибута он не должен использоваться в классификаторах, для этого нужно убрать флажок классификатора в «Свойствах атрибута», после чего появится возможность его удалить.

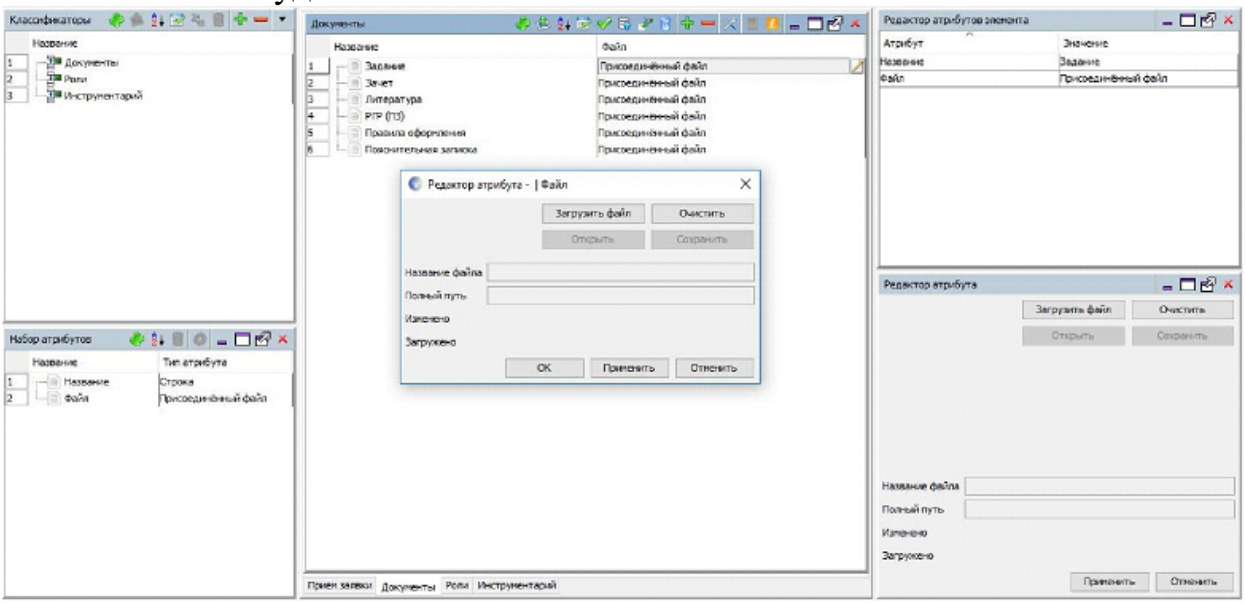

Рисунок 9 – Диалоговое окно «Редактор атрибута»

Также в программе имеется возможность присоединять элементы классификатора к названиям стрелок. Для этого необходимо открыть диалоговое окно «Параметры стрелки» в рабочем пространстве «Диаграммы», дважды кликнув по стрелке. Для присоединения элементов классификатора нужно нажать кнопку «Добавить» и выбрать элемент классификатора или весь классификатор, если это необходимо (рис. 10). Можно присоединять несколько элементов классификатора, тогда они будут указаны на диаграмме через точку с запятой. Удалить их можно с помощью кнопки «Удалить».

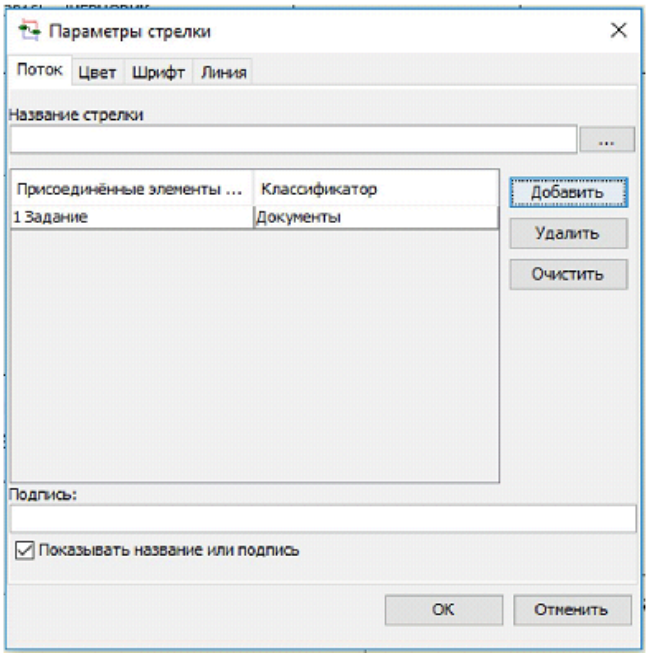

Рисунок 2.10 – Диалоговое окно «Параметры стрелки»

IDEF0-модель сохраняется на диске с помощью команд «Файл» «Сохранить» или «Файл» «Сохранить как…».

Далее приведен пример IDEF0 модели «Выполнить расчетно-графическую работу (РГР)» (рис. 11, 12).

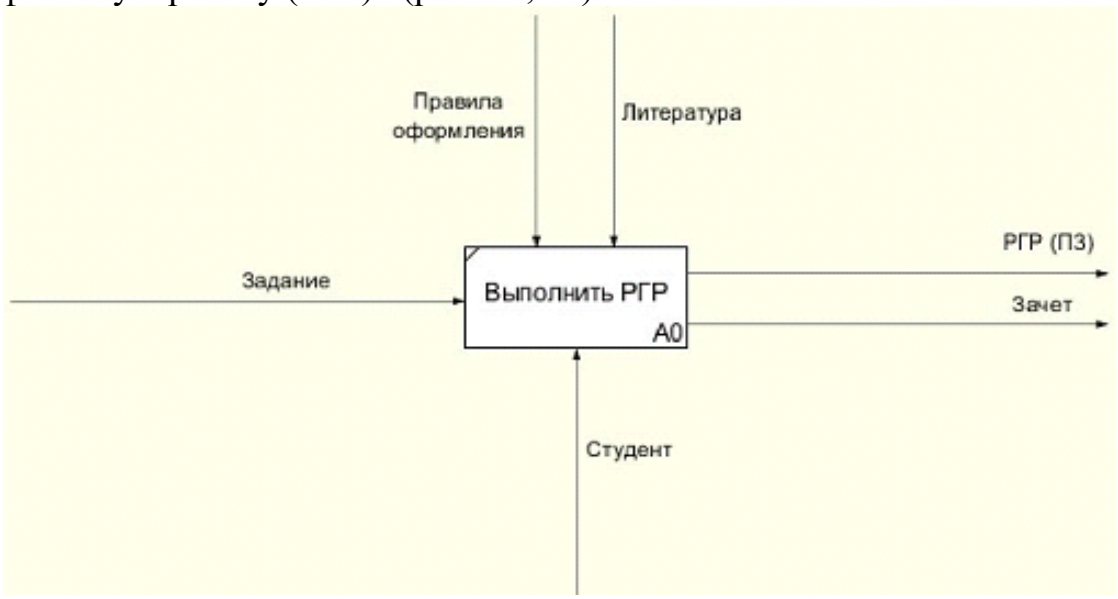

Рисунок 2.11 – Первый уровень функциональной модели «Выполнить РГР»

Элементы классификатора функциональной модели «Выполнить РГР»:

 Задание – постановка задачи на выполнение РГР.

 Зачет – оценка в баллах, выставляемая студенту за РГР.

 Литература – список литературы для самостоятельной проработки в рамках РГР.

 Метод решения – метод решения поставленной задачи.

 План, график работы – план выполнения РГР.

 Пояснительная записка – оформленная в соответствии с требованиями пояснительная записка к РГР.

 Правила оформления – методические указания по выполнению и оформлению РГР.

 РГР (ПЗ) – расчетно-графическая работа (пояснительная записка).

 Расчеты – расчеты, необходимые для решения поставленной задачи.

 Студент – учащийся ВУЗа.

Иерархическое представление функциональной модели «Выполнить РГР»:

[A0] Выполнить РГР

[A1] Составить план работы

[A2] Изучить литературу и выбрать метод (методику, приемы решения)
[A3] Выполнить необходимые расчеты на ЭВМ

[A4] Подготовить пояснительную записку

[A5] Защитить РГР

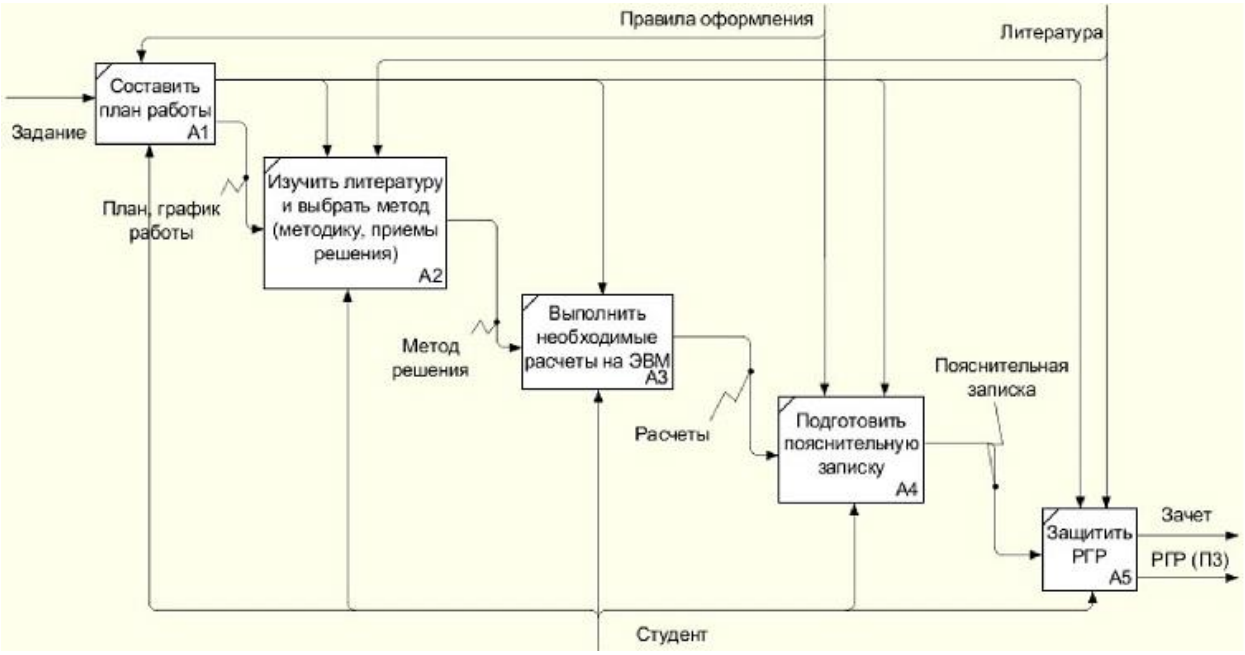

Рисунок 2.12 – Декомпозиция блока A0 функциональной модели «Выполнить РГР»

2 Создание словаря данных модели.

Перейти в рабочее пространство «Классификаторы». В работе будет создано два типа классификатора: Функция и Кадры. В классификаторе «Кадры» будут содержаться элементы для описания подразделений персонала, таких как «Отдел приема заявок», «Цех» и «Отдел доставки».

После их создания, для классификатора «Кадры» нужно создать атрибуты, в которых будет храниться информация о подразделении (руководитель, количество сотрудников, списочный состав, место расположения, телефон). Для этого в окне «Набор атрибутов» нажать кнопку и в появившемся диалоговом окне «Создать атрибут» дать присвоить название атрибуту и соответствующий тип и потом поставьте флажок у классификатора «Кадры». Аналогично требуется создать следующие атрибуты для данного классификатора:

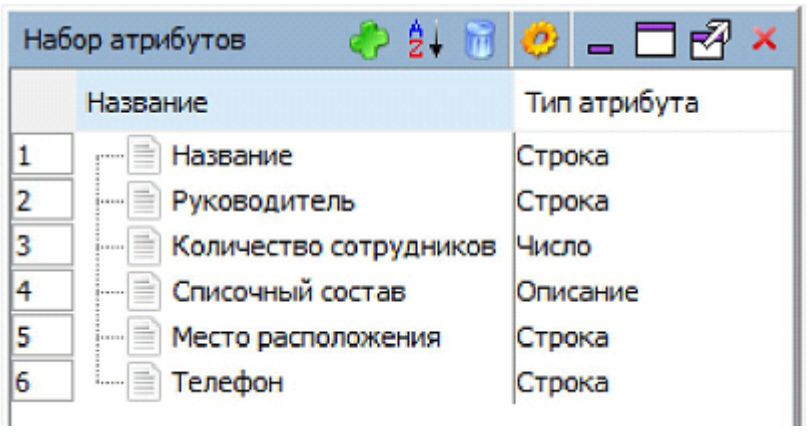

Рисунок 13 – Окно «Набор атрибутов»

После этого в классификаторе «Кадры» появятся дополнительные поля:

Руководитель, Количество сотрудников, Списочный состав, Место расположения, Телефон (рис. 14). Их нужно заполнить для каждого отдела. Нужно обратить внимание на то, чтобы заполнить «Списочный состав» с типом атрибута: описание, необходимо нажать на кнопку справа от описания или ввести текст в окне «Редактор атрибута».

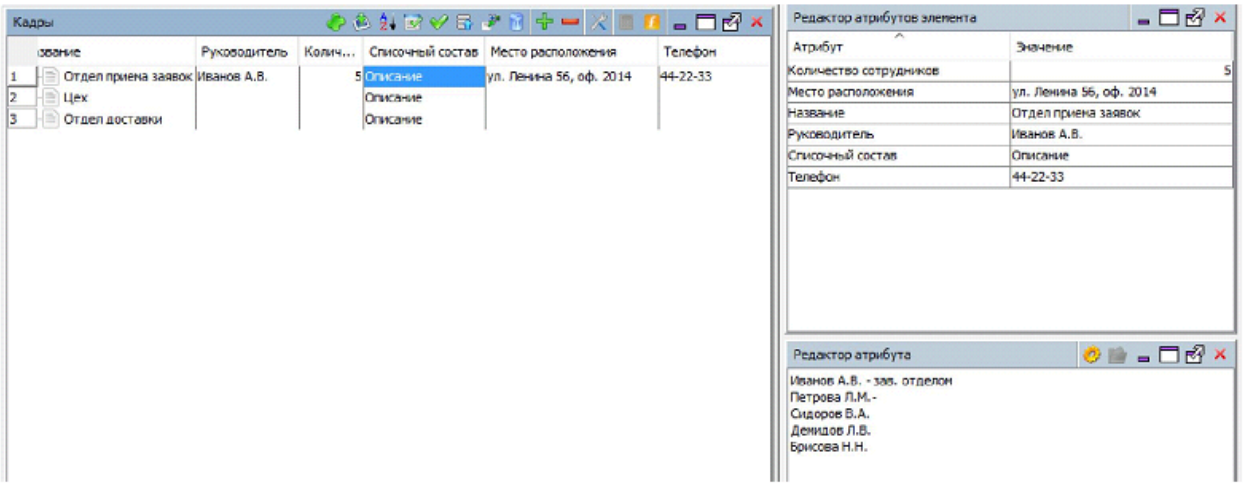

Рисунок 14 – Рабочее пространство «Классификаторы»

После этого нужно перейти к созданию классификатора «Функции». Для этого в центральном окне рабочего пространства нужно создать элементы классификатора в соответствии с названиями функций на диаграмме: «Прием заявки», «Изготовление продукта» и «Доставка продукта». Этот классификатор будет использоваться для описания функциональных блоков модели, и будет содержать информацию о соответствующей функции (описание функции, средняя продолжительность ее выполнения, сотрудники, выполняющие функцию).

Аналогично созданию атрибутов для предыдущего классификатора, требуется создать три атрибута, не забывая при создании ставить флажок для классификатора «Функции»:

- описание – тип: описание;
- продолжительность (час) – тип: число;
- сотрудники – тип: элемент классификатора (установите классификатор «Кадры», созданный Вами раннее и его атрибут «Название»).

После этого нужно заполните появившиеся поля. Для «Приема заявок» в поле «Описание» ввести, например, такой текст: «Консультирование клиента. Оформление заказа. Прием оплаты.», ввести значение для поля «Продолжительность (час)» - 0.5. Следует обратить внимание на то, что в поле «Сотрудники» появляется выпадающий список, в котором необходимо выбрать соответственно «Отдел приема заявок» (рис. 15). Аналогично заполняются оставшиеся элементы классификатора.

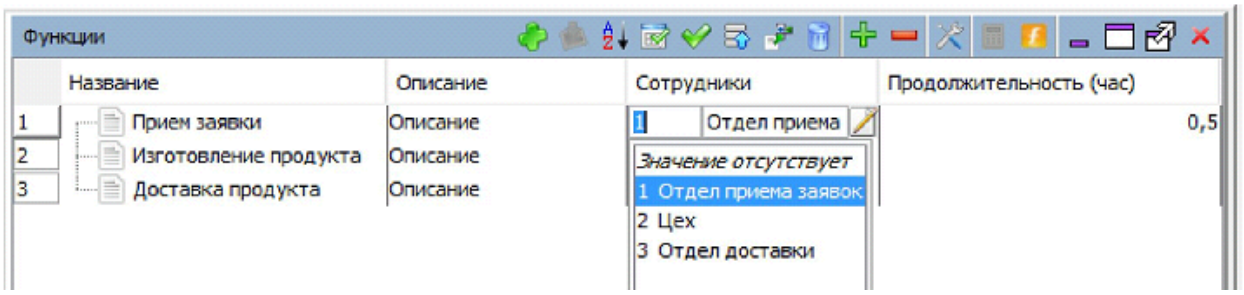

Рисунок 15 – Окно задания значений полей записи

Для обновления номера версии модели нужно выбрать команду меню «Файл» – «Сохранить».

Созданные классификаторы можно экспортировать и импортировать. Для экспорта выбрать «Файл» – «Экспортировать классификаторы». Выбрав путь сохранения, в появившемся диалоговом окне «Классификаторы», нужно указать те, которые необходимо экспортировать, после чего нажать ОК. Классификаторы сохраняются в том же формате .rsf, что и проект.

Для выбранного в качестве индивидуального задания бизнес-процесса нужно выделить основные типы объектов, информация о которых будет содержаться в словаре. Это могут быть:

- подразделения организации, участвующие в выполнении бизнес-процесса;
- документы (планы, заказы, проекты), создаваемые в ходе выполнения процесса;
- продукция (услуги), создаваемые в ходе выполнения процесса;
- этапы (подэтапы, работы) процесса.

Для каждого выделенного типа объектов требуется составить список атрибутов (полей данных), достаточных для описания данного типа. Также нужно определить какого типа должны быть атрибуты.

Далее для каждого типа объектов нужно составить перечень конкретных объектов данного типа (содержащихся в модели в качестве меток или блоков). Затем требуется определить какими должны быть конкретные значения атрибутов для каждого из выделенных объектов.

После этого нужно определить типы записей. Каждому типу объектов должен соответствовать свой тип записей. При определении типа записи каждому атрибуту сопоставляется поле данных. С учетом этого нужно создать записи для конкретных объектов, содержащихся в модели в качестве меток или блоков. При создании записи ввести значения атрибутов в соответствующие поля данных. После этого требуется сохранить словарь.

3 Дополнение модели DFD диаграммами

Диаграммы потоков данных (DFD) – представляют собой сеть связанных между собой работ. Их удобно использовать для описания документооборота и обработки информации. DFD описывает:

- функции обработки информации – работы;

 документы, объекты, сотрудников или отделы, участвующие в процессе обработки информации;

 внешние ссылки, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;

 таблицы для хранения документов (хранилища данных).

Для построения DFD-диаграммы используются поток данных, процесс, хранилище и внешняя сущность.

Потоки данных являются механизмами, использующимися для моделирования передачи информации (или физических компонентов) из одной части системы в другую. Потоки изображаются на схеме именованными стрелками, ориентация которых указывает направление движения информации. Стрелки могут подходить к любой грани работы и могут быть двунаправленными для описания взаимодействия типа команда-ответ.

Назначение процесса состоит в продуцировании выходных потоков из входных, в соответствии с действием, задаваемым именем процесса. Каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы.

Хранилище данных позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилища – это «срезы» потоков данных во времени. Информация, содержащаяся в хранилище, может использоваться в любое время после её определения при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое.

Внешняя сущность представляет собой сущность вне контекста системы, являющуюся источником или приемником данной системы. Предполагается, что объекты, представленные внешними сущностями не должны участвовать ни в какой обработке. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

Для знакомства с процессом создания DFD-диаграмм, нужно открыть модель «Создание продукта», созданную на предыдущем занятии. Далее выбрать работу «Оформление заказа» и построить дочернюю диаграмму в нотации DFD (рис. 16).

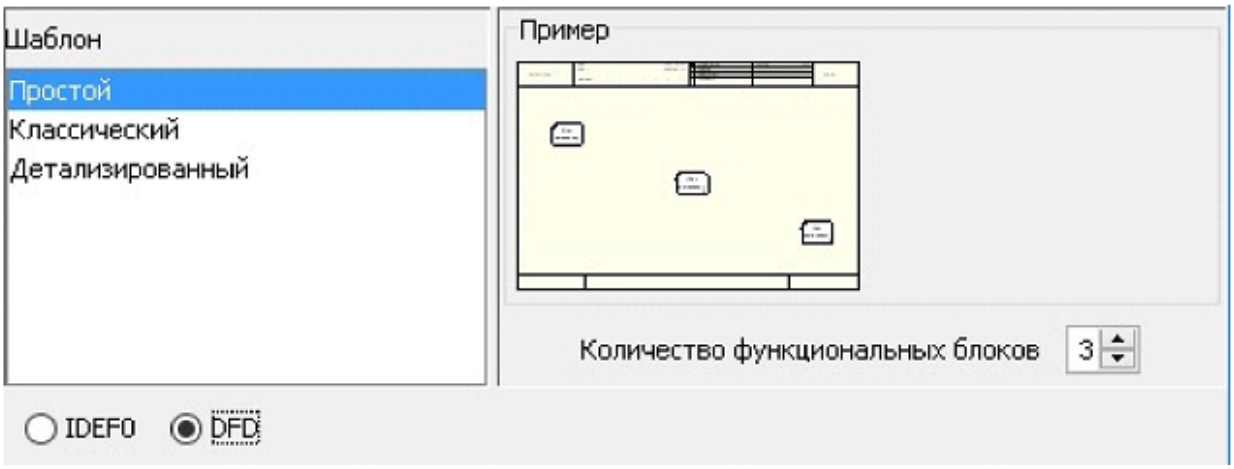

Рисунок 16 – Создание DFD-диаграммы

При формировании декомпозиции нужно внести на диаграмму имена работ:

«Проверка и внесение клиента», «Заполнение формы заказа», «Формирование заявки на доставку».

Согласно нотации DFD диаграмма не должна иметь граничных стрелок-все стрелки должны начинаться и заканчиваться на работах, хранилищах данных или внешних сущностях. Поэтому, если строго следовать правилам нотации, следует: удалить все граничные стрелки на диаграмме DFD. И создать соответствующие внешние сущности и хранилища данных.

Для этого нужно добавить на диаграмму внешнюю сущность, выбрав соответствующую кнопку на панели инструментов , затем сделать по объекту двойной щелчок и в появившемся диалоговом окне выбрать «Задать DFD объект». Затем в появившемся диалоговом окне в контекстном меню выбрать пункт «Создать элемент». Название классификатора «Внешние сущности» можно ввести в созданную строку, дважды, медленно кликнув мышью по строке, или же нажав клавишу F2, предварительно выделив нужную строку мышью. Затем нужно добавить к созданному классификатору элемент «Клиент» (рис. 17). Аналогично создается внешняя сущность «Поставщики».

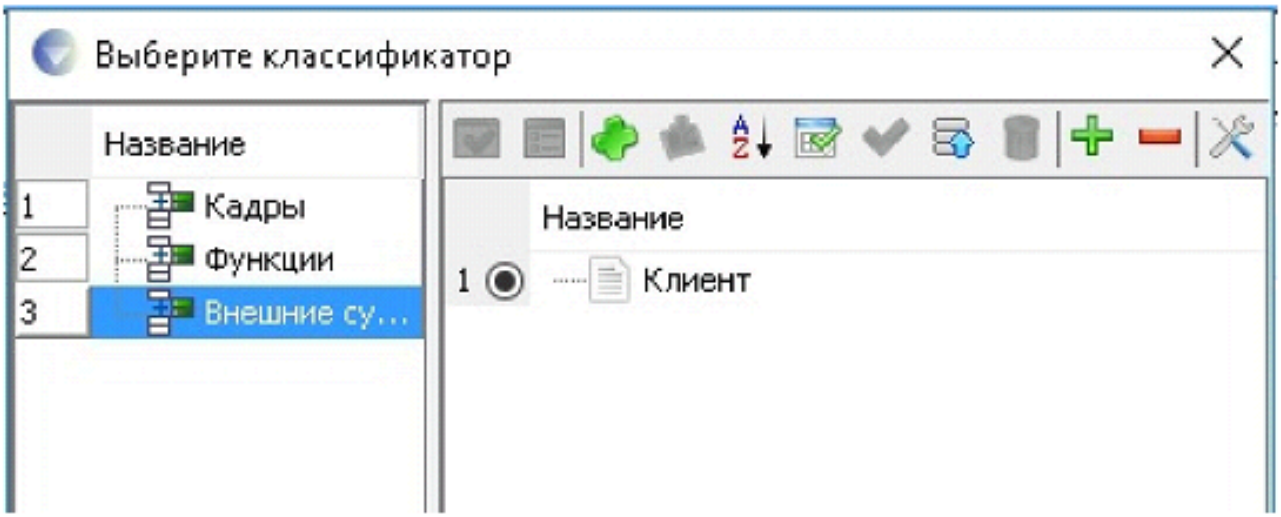

Рисунок 17 – Создание элемента Клиент классификатора Внешние сущности

Для внесения в модель хранилища используется кнопка . Затем нужно создать классификатор «Хранилища данных» и элемент для него «Заказы». Аналогично задаются элементы «Клиенты» и «Заявки на доставку». После этого нужно создать внутренние стрелки и именовать их (рис. 18).После чего сохранить модель.

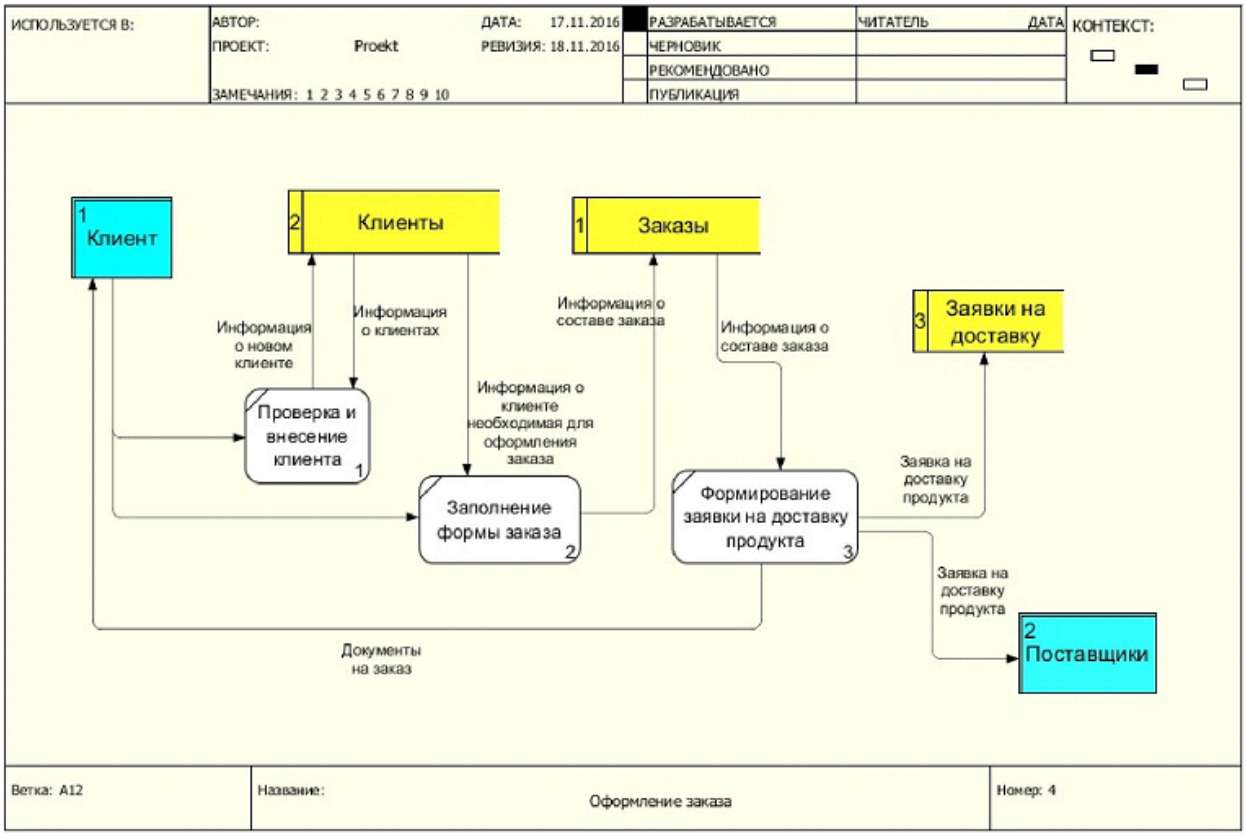

Рисунок 18 – DFD-диаграмма декомпозиции процесса оформления заказа

Требуется создать несколько DFD диаграмм для работ описывающих обработку информации, документооборот.

Отчет по работе должен содержать

1. Номер, название и цель работы
2. Описание процессов для диаграмм IDEF0 (сами диаграммы - в приложении к отчету:
   контекстная,
   детализации, содержащая не менее 4 процессов,
   декомпозиции 2 уровня – 2 диаграммы)
3. Классификатор модели
4. Описание процессов для диаграмм DFD (сами диаграммы - в приложении к отчету: декомпозиции 2 уровня – 2 диаграммы)
5. Вывод
6. Файл * .rsf

Контрольные вопросы

1. Основные элементы диаграмм IDEF0
2. Основные элементы диаграмм DFD
3. Назначение и порядок разработки классификаторов