

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 10.10.2021 20:56:54
Уникальный программный ключ:
bb52f959411e611177f6e9977b97e87139b1a9d



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Колледж экономики, управления и права

**Методические указания
по организации практических занятий
по учебной дисциплине
Бизнес-информатика**

Специальность

09.02.04 Информационные системы (по отраслям)

Ростов-на-Дону

2021

Методические указания по разработаны с учетом ФГОС среднего профессионального образования специальности 09.02.04 Информационные системы (по отраслям) предназначены для студентов и преподавателей колледжа.

Методические указания определяют этапы выполнения работы на практическом занятии, содержат рекомендации по выполнению практических занятий, а также содержит список рекомендованной литературы.

Составитель (автор): преподаватель колледжа ЭУП Маловечко С.Н.

Рассмотрены на заседании предметной (цикловой) комиссии специальности 09.02.04 Информационные системы (по отраслям)

Протокол № 7 от 30 мая 2021 г

Председатель П(Ц)К специальности  С.В. Шинакова

личная подпись

и одобрены решением учебно-методического совета колледжа.

Протокол 6 от 31 мая 2021 г

Председатель учебно-методического совета колледжа  С.В. Шинакова

Рекомендованы к практическому применению в образовательном процессе.

Практическая работа 1

Знакомство с программой Bizagi Process Modeler

1. Цель работы

Целью работы является изучение технологии построения модели процесса с использованием Bizagi Process Modeler,

2. Задачи

Основными задачами лабораторного практикума являются: закрепление и расширение теоретических знаний студентов, приобретение навыков моделирования простых процессов в Bizagi Process Modeler.

3. Теоретическая часть

Bizagi Process Modeler это очень эффективная программа, специализирующаяся на разработке диаграмм или цепей бизнес-процессов.

Программный продукт Bizagi позволяет строить модели деятельности, выполнять и совершенствовать бизнес-процессы, используя графическую среду, без необходимости программирования.

Моделирование в BPMN 2.0 осуществляется посредством диаграмм с небольшим числом графических элементов. Это помогает пользователям быстро понимать логику процесса. Выделяют четыре основные категории элементов:

- Объекты потока управления: события, действия и логические операторы
- Соединяющие объекты: поток управления, поток сообщений и ассоциации
- Роли: пулы и дорожки
- Артефакты: данные, трупы и текстовые аннотации.

Элементы этих четырёх категорий позволяют строить простейшие диаграммы бизнес-процессов. Для повышения выразительности модели спецификация разрешает создавать новые чипы объектов потока управления и артефактов.

Объекты потока управления

Объекты потока управления разделяются на три основных типа: события {events}, действия (activities) и логические операторы (gateways).

События изображаются окружностью и означают какое-либо происшествие в мире. События инициируют действия или являются их результатами. Согласно расположению, в процессе события могут быть классифицированы на начальные (start), промежуточные (intermediate) и завершающие (end).



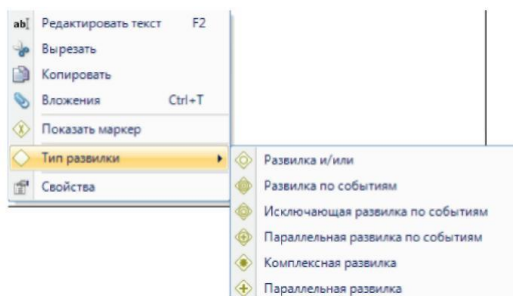
Рис.3.1 Старт процесса, событие, завершение



Рис.3.2 Действия

Действия изображаются прямоугольниками со скругленными углами. Среди действий различают задания и подпроцессы. Графическое изображение свернутого подпроцесса снабжено знаком плюс у нижней границы прямоугольника.

Логические операторы изображаются ромбами и представляют точки принятия решений в процессе. С помощью логических операторов организуется ветвление и синхронизация потоков



управления в модели процесса

Рис.3.3 Типы логических операторов в bizagi

* Оператор *исключающего «или»*, управляемый данными ([англ. data-based exclusive gateway](#)). Если оператор используется для ветвления, то поток управления направляется лишь по одной исходящей ветви. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток.

* Оператор *исключающего «или»*, управляемый событиями ([англ. event-based exclusive gateway](#)) направляет поток управления лишь по той исходящей ветви, на которой первой произошло событие. После оператора данного типа могут следовать только события или действия-обработчики сообщений.

* Оператор *включающего «или»* ([англ. inclusive gateway](#)) активирует одну или более исходящих ветвей, в случае, когда осуществляется ветвление. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток,

* Оператор «и» ([англ. parallel gateway](#)), использующийся для ветвления, разделяет один поток управления на несколько параллельных. При этом все исходящие ветви активируются одновременно. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех входящих ветвей и лишь затем активирует выходной поток.

* Сложный оператор ([англ. complex gateway](#)) имеет несколько условий, в зависимости от выполнения которых активируются исходящие ветви. Оператор затрудняет понимание диаграммы, так как условия, определяющие семантику оператора, графически не выражены на диаграмме. Вследствие этого использование оператора нежелательно.

Соединяющие объекты

Объекты потока управления связаны друг с другом соединяющими объектами. Существует три вида соединяющих объектов: потоки управления, потоки сообщений и ассоциации.

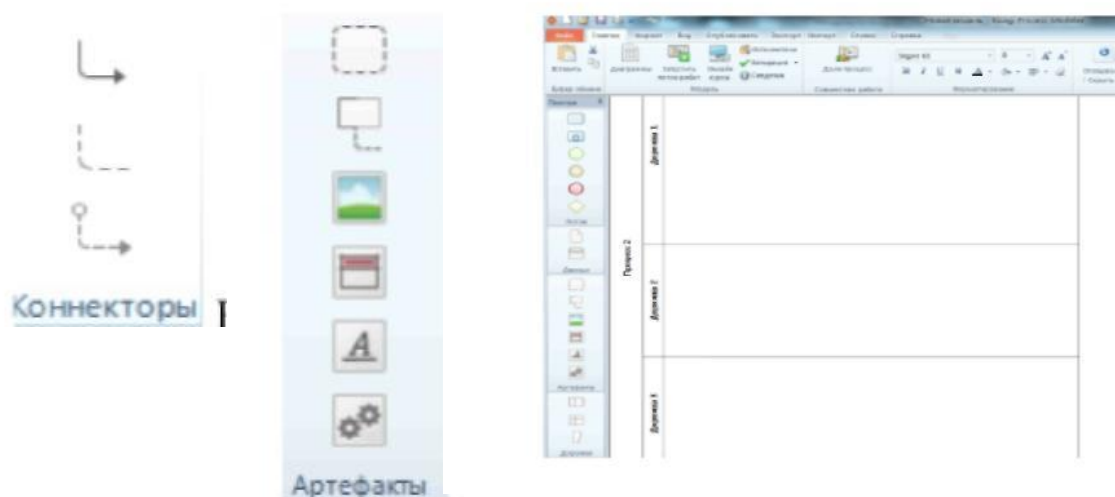
Поток **управления** изображается сплошной линией, оканчивающейся закрашенной стрелкой. Поток управления задаёт порядок выполнения действий. Если **линия** потока управления перечеркнута диагональной чертой со стороны узла из которого она исходит, то она обозначает поток, выполняемый по умолчанию. (При правильном соединении с помощью потока управления действие должно выделиться зеленым цветом.)

Ассоциации изображаются пунктирной линией, заканчивающейся стрелкой. Ассоциации используются для ассоциирования артефактов, данным или текстовых аннотаций с объектами потока управления,

Поток сообщений изображается штриховой линией, оканчивающейся открытой стрелкой. Поток сообщений показывает какими сообщениями обмениваются участники. Поток сообщений изображает взаимодействие между двумя сущностями, готовыми к их отправке и получению. В BPMN два отдельных пула на диаграмме обозначают две сущности.

Роли — визуальный механизм организации различных действий в категории со сходной функциональностью. Существует два типа ролей:

Пулы изображаются прямоугольником, который содержит несколько объектов потока управления, соединяющих объекты и артефакты. Пул представляет участника процесса. Участником может быть любая организация, например, компания) или бизнес - роль (например, покупатель, производитель, продавец)



Дорожки представляют собой часть пула. Дорожки позволяют организовать объекты потока Пулуправления, связывающие объекты и артефакты.

Артефакты позволяют разработчикам отображать дополнительную информацию в диаграмме. Это делает диаграмму более удобочитаемой и насыщенной информацией. В bizagi существуют несколько видов артефактов. Группа, аннотация, изображение, заголовок, форматированный текст, пользовательские артефакты

Практическая работа 2

Создание проекта в BizagiProcessModeler

1. Цель работы

Целью работы является изучение технологии построения модели процесса с использованием BizagiProcessModeler,

2. Задачи

Основными задачами лабораторного практикума являются: закрепление и расширение теоретических знаний студентов, приобретение навыков моделирования простых процессов в BizagiProcessModeler.

3. Практическая часть

1. Шаг 1 Запуск программы Bizagi с рабочего стола. Непосредственно после открытия Bizagi отображается страница:

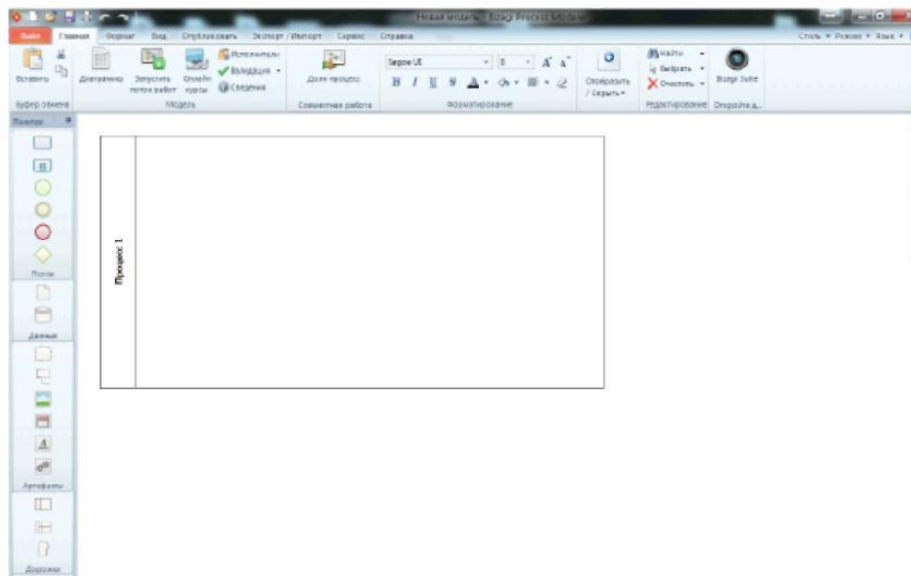


Рис. 1 Начальная страница BizagiProcessModeler

2. Шаг 2- создание проекта. В меню программы перейти на вкладку *Файл > новый > создать пустую модель*, после чего запустится процесс создания нового проекта с пулом. (Например модель будет называться "Протокол совещания при управляющем директоре") Для этого нужно кликнуть левой клавишей мыши по "Процесс 1">*редактировать текст*>*название проекта*.

В нашем случае процесс проведения совещания при управляющем директоре на предприятии выглядит следующим образом: *протокол совещания проходит по регламенту, на нем решается, ставить или нет пункты на контроль исполнения. Далее начальнику сектора протокола следует оформить протокол совещания в бумажном виде, подписать у управляющего директора и совершить рассылку копий для ознакомления в другие подразделения предприятия.*

В этом процессе будут задействованы следующие роли:

- -управляющий директор
- -секретарь-референт
- -начальник сектора протокола


- -подразделение предприятия




Протокол совещания	Управляющий директор	
	Секретарь-референт	
	Сектор протокола	
	Подразделение предприятия	

Рис.2 - Пул

1.Для того, чтобы создать роль, нужно добавить дорожку  в пул. и переименовать в название выбранной роли.

2.Старт бизнес-процесса  (зеленый круг на панели инструментов)


3. Далее нужно расписать бизнес - процесс по задачам. Задача должна представлять собой элементарное действие в рамках процесса. На панели инструментов выбираем компонент задача  и перетаскиваем в нужную дорожку. Для переименования следует кликнуть правой кнопкой мыши > *редактировать текст*.

В примере для роли управляющего директора предусмотрены следующие задачи: "провести совещание", "поставить пункты на контроль", если было принято это решение. Для того, чтобы указать в модели выбор решения да/нет следует сделать развилку. На панели инструментов выбрать компонент развилка  затем указать маркер,  и переименовать "Есть пункты на контроль?" если имеются, то следует задача "поставить пункты на контроль", если нет, то следует параллельная развилка к задаче "подготовить протокол в MSWord". В любом случае, есть пункты на контроль или нет, сектору протокола следует оформить протокол в бумажном виде. Для этого нужно использовать параллельную развилку:  для синхронизации параллельных потоков операций.

4. Задачи на дорожке сектора протокола: "подготовить протокол в MSWord", "распечатать протокол, поставить штамп "проверено сектор протокола", после того, как на нем будет стоять подпись директора (задача "передать на подпись УД"), следует "получить и отсканировать бумажный документ", затем "сформировать список рассылки протокола" и разослать по подразделениям предприятия - "отправить протокол в соответствии со списком рассылки".

5. Задачи секретаря-референта: получить/отправить в нужное место протокол в бумажном виде (задачи "получить бумажный документ, передать на подпись УД", и "отправить подписанный протокол в сектор протокола").

6. Задачи роли подразделения предприятия - "получить документ" и "поставить пункты на контроль"

7. Завершение  обозначает завершение потока управления в рамках процесса (при этом другие потоки могут продолжать исполнение)

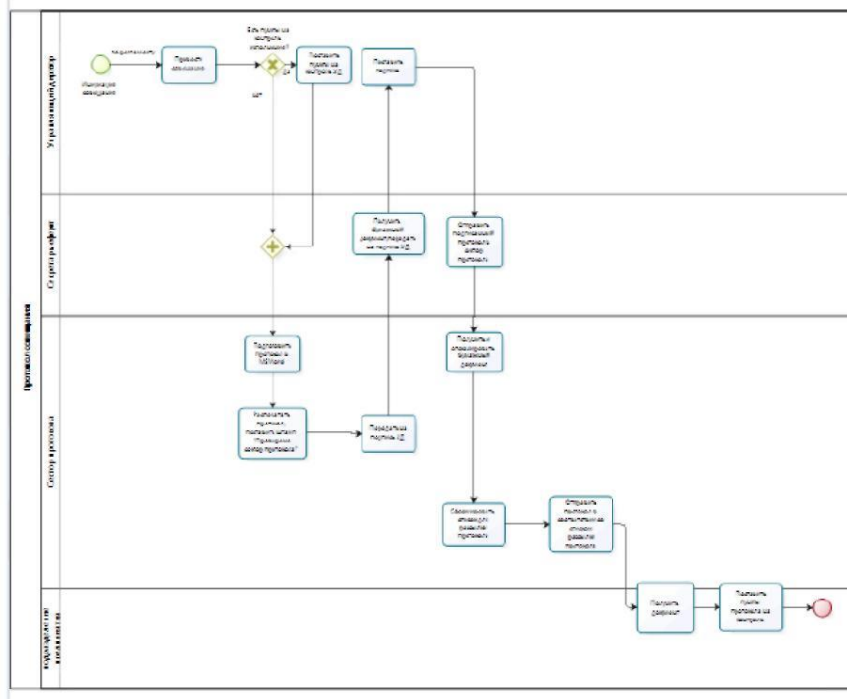


Рис.3 - Схема процесса.

8. Для того, чтобы декомпозировать модель, следует выбрать нужный блок (задачу) диаграммы. Выберем первый блок "провести совещание" правой клавишей мыши выбираем функцию "преобразовать в подпроцесс", после чего внизу появляется новая вкладка провести совещание. Графическое изображение свернутого подпроцесса снабжено знаком плюс у нижней границы прямоугольника.



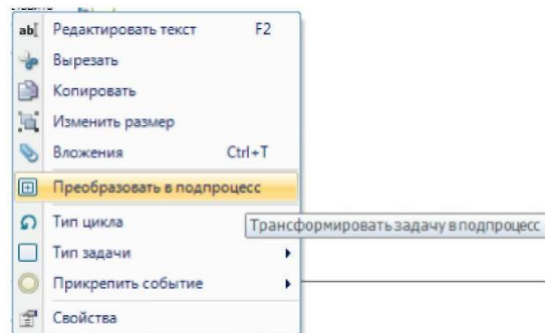
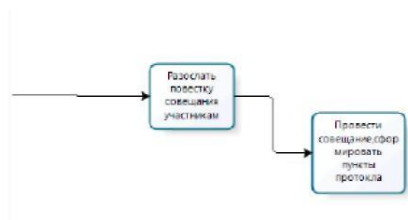


Рис. 4 Преобразование задачи в подпроцесс



Далее составляем задачи подпроцесса

Рис. 5 Подпроцесс задачи

В итоге бизнес-процесс "Протокол совещания примет вид"

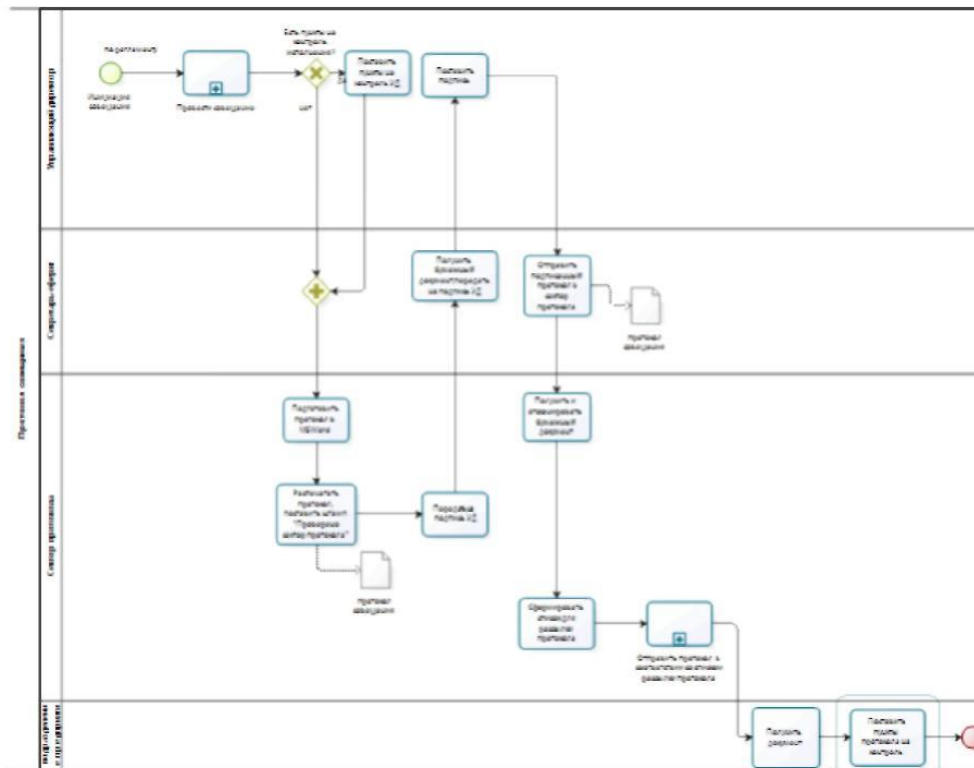


Рис.6 Схема процесса с подпроцессами

Для составления отчетов бизнес -процесса нужно перейти во вкладку "опубликовать" на панели управления и выбрать нужный вариант:

- Word
- Pdf

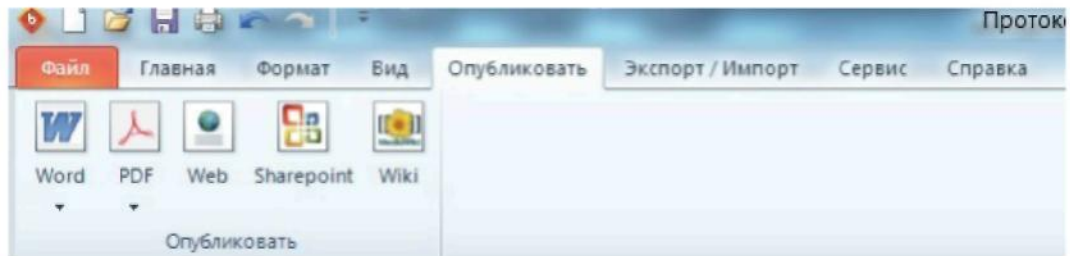
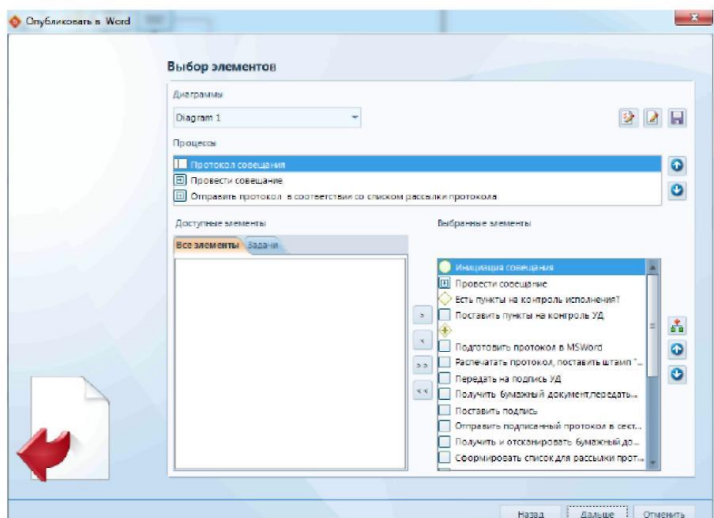


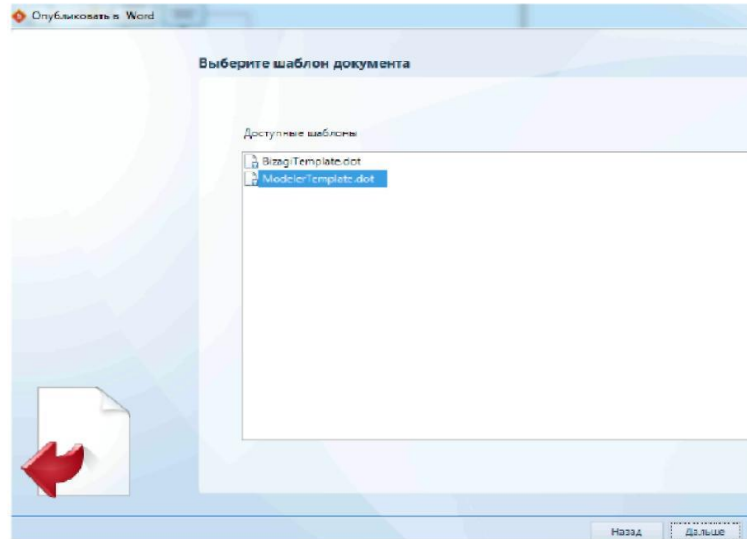
Рис.7. Выбор варианта отчета

-Web

Выберем вариант Word, откроется окно "опубликовать в Word"Выбираем Diagram1, нажимаем кнопку "далее", выбираем все доступные компоненты:

Рис. 8 - Выбор элементов





Выбор шаблона документа следует выбрать ModelerTemplate, dot

Рис. 4.9 - Выбор шаблона документа

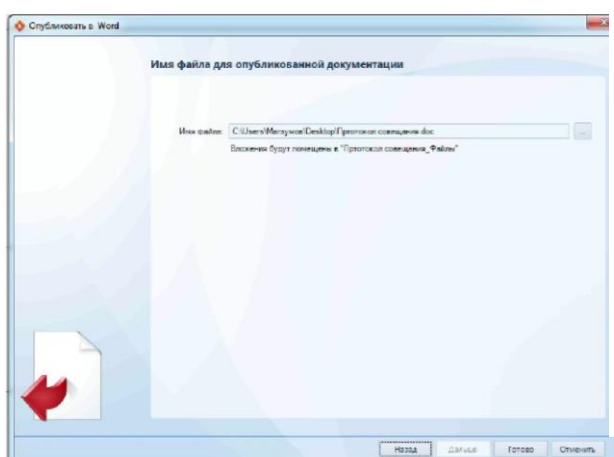


Рис.10 Имя файла для опубликованной документации

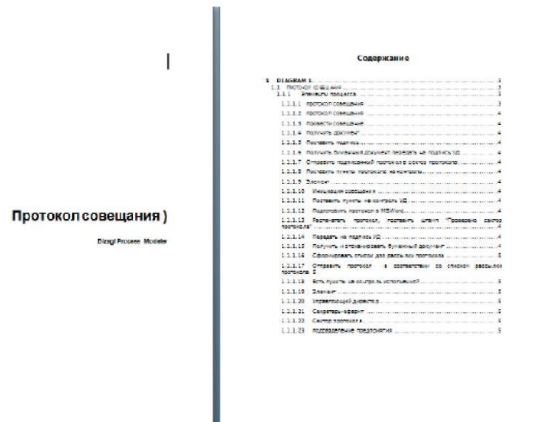
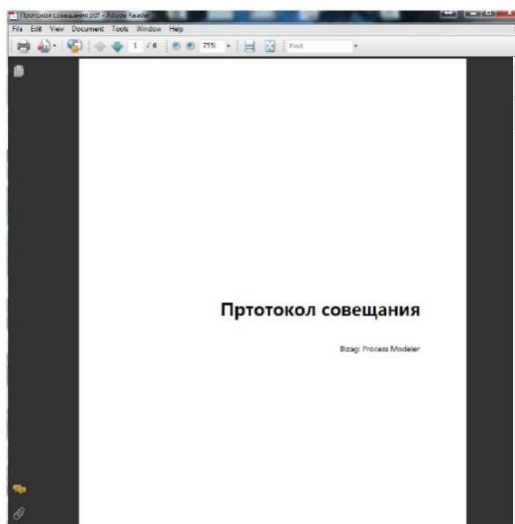


Рис.11 публикация в MSWord



Аналогичным образом создадим отчеты в форматах pdf и в web. :

Рис. 12 - публикация в pdf

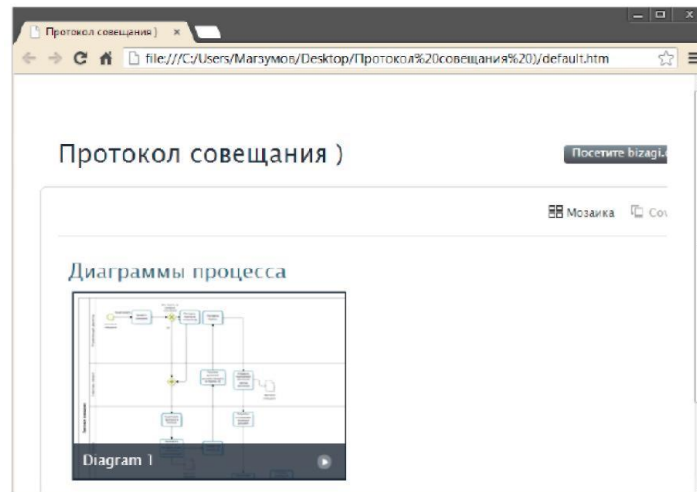


Рис. 13 - публикация в web

Практические занятия № 3

Создание новой модели и процесса в BizagiModeler

1. Цель работы

Целью работы является изучение технологии построения модели процесса с использованием BizagiProcessModeler,

2. Задачи

Основными задачами лабораторного практикума являются: закрепление и расширение теоретических знаний студентов, приобретение навыков моделирования простых процессов в BizagiProcessModeler.

Процесс будет называться Zakaz_Avto (Заказ автотранспорта).

Легенда следующая: есть компания, у неё некоторый парк собственных автомобилей и потенциально любой сотрудник компании может заказать автомобиль, для каких то служебных нужд. Соответственно проект будет назван Zakaz_Avto.


Заказчик просит автомобиль, после этого он попадает в канцелярию, где эту заявку одобряют или нет, после этого гараж окончательно удовлетворяет заявку.



Здесь будут три действующих лица:

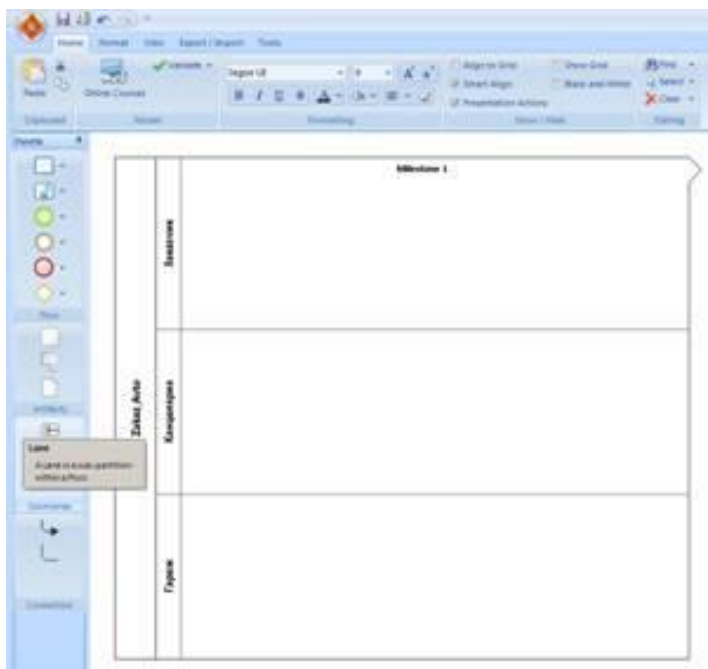
1. Заказчик


2. Канцелярия

3. Гараж


1) Для указания действующих лиц используется компонент **Lane**  его необходимо прикрепить к уже существующему и задать имя (необходимо в Properties (свойства), в пункте DisplayName указать его имя).

2) Старт бизнес процесса **Start Event**  (Зелёный круг на панели инструментов). Следующий шаг можно произвести двумя способами: Выбрать на панели инструментов компонент **Task** и установить его, либо кликнуть на установленный компонент **Start Event** (там будут доступны 3 компонента для дальнейших действий) удерживая левую кнопку мыши на компоненте **Task**  перетащить его в нужное место. Первый шаг бизнес процесса будет называться «**Ввести заявку**» (необходимо в Properties (свойства) в пункте DisplayName указать его имя).



3) Заявка попадает в канцелярию, которая должна принять решение одобрить заявку или нет. Необходимо создать компонент **Task** с именем «**Добавить заявку**» на дорожке «Канцелярия». Так же на этой дорожке необходимо разместить шлюз (Gateway)  именем которого будет «**Одобрено?**»

4) Если одобрено, то процесс идет дальше в гараж, шаг с названием «**Принять заявку**», за которой так же аналогичный шлюз, но уже с именем «**Принято**».

5) Если не одобрено, переход к задаче «**Машина не выделена**». На этом процесс заканчивается конечным событием (End event)  «**Отказ**».

Если же отказ произошел на шлюзе с именем «**Принято?**» то происходит возврат потока операции на задачу «**Одобрить заявку**», потому что возможен сценарий, когда машины на указанную дату нет, или нет указанного типа, но есть на другую дату, чтобы окончательно отказывала всегда канцелярия.

б) Для этого необходимо нажать на шлюз «**Принято**» выбрать **Task** и перенести его в задачу «**Одобрить заявку**», таким образом, произойдет соединение шлюза и задачи. Это в случае нет.



7) В случае да, необходимо поставить *параллельный шлюз*. Параллельный шлюз представляет собой механизм для синхронизации параллельных Поточков операций. Для этого необходимо *правой кнопкой кликнуть на нужный шлюз найти Gateway type и выбрать пункт Parallel Gateway*.

8) Далее Задача «**Машина выделена**» на дорожке **Заказчик**. И поставить задачу водителю выполнить рейс, которая будет находиться на дорожке **Гараж**.

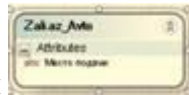
9) Так же необходимо объединить эти потоки операций для этого необходимо поставить ещё одну параллельный шлюз (**Parallel Gateway**). На этом процесс заканчивается конечным событием (End event) «**Успех**».

10) Так же можно поставить *нормативную продолжительность задач* (свойства задачи (**Properties**) внизу есть пункт **Duration**: дни, часы, минуты), указать по 1 часу для каждой из задач.



1) Далее к следующему шагу, **Model data** (моделирование данных). *Выбрать пункт **Model data**. Система просит указать **Process Entity**, выбрать **Zakaz_Avto**.*

У Vizagi для хранения атрибутов реализована очень простая, но крайне удачная концепция. *Каждому процессу соответствует своя таблица базы данных, соответственно одному экземпляру процесса соответствует одна запись. С развитием появятся другие таблицы. Процессная сущность это физическая таблица в базе данных.*



Пока что существует 1 таблица именем **Zakaz_Avto**, которого является название процесса **Zakaz_Avto**.

Display Name	Name	Type	State
Место подачи	Mesto podachi	String	

Description: Default Value:
Help Text: Length: 50

<< Back Next >> Finish Cancel

Рисунок 4 – Лист атрибутов

2) Необходимо задать атрибуты, которые будут столбцами этой таблицы:

1. Место подачи
2. Время подачи
3. Маршрут
4. Пассажиры

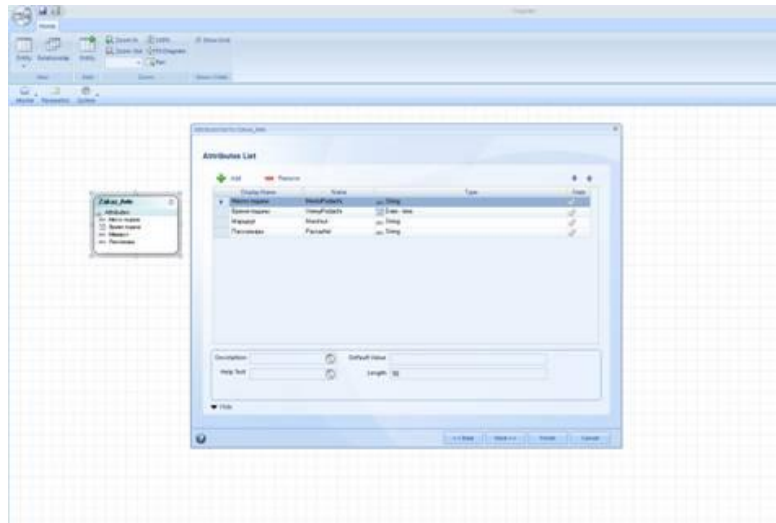


Рисунок 5 – Modeldata

Для этого необходимо нажать правой кнопкой мыши на сущность **Zakaz_Avto** и выбрать там пункт **Edit Attribute List**, после чего появится окно, для заполнения атрибутов состоящее из нескольких шагов. Для каждого атрибута необходимо заполнить 3 поля: **Display Name, Name, Type**.

Здесь и в других местах (в имени Задачи и в имени проекта) у Bizagi для элементов есть два имени **Name** и **Display Name**. На **Name** достаточно жесткие ограничения, потому что это имя транслируется в имена переменных, не допускается кириллица.

С другой стороны в **Display Name** можно задавать имена на нескольких языках.

Тип определяет, какого рода данные будут храниться в атрибуте. Кроме стандартных (**String, Date Time, Boolean**), здесь можно указать и более сложный тип или создать новый.

3) После заполнения всех полей нажать **Finish**.

5.3 Создание форм

1) Далее к следующему шагу, **Define Forms** (Экранные формы) .

Необходимо обратить внимание, что сейчас *все шаги процесса выполняются людьми. Пользователь компьютера получает задачу и каждой этой задачи должна быть определена экранная форма.*

На рисунке 6 Показана схема и *желтым восклицательным знаком* показаны задачи, для которых эта форма не определена. Далее нужно кликнуть на задачу и определить эту форму.

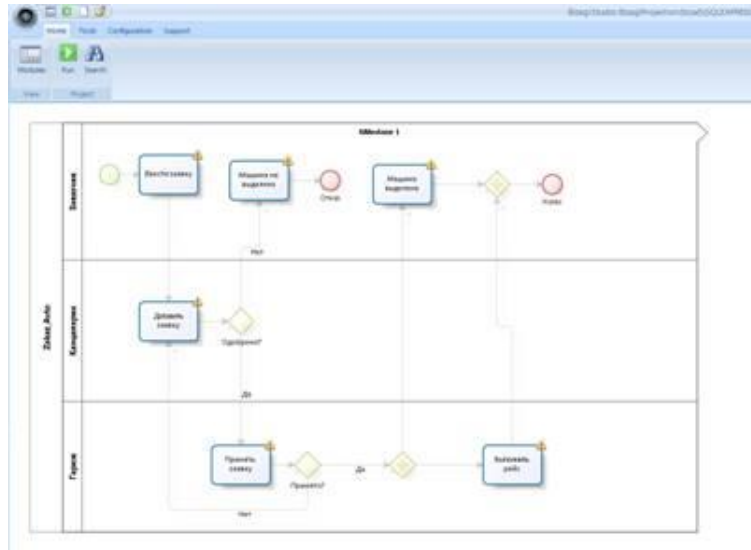


Рисунок – 6 - **Define Forms**

Слева модель данных, справа рабочее поле, на котором будут располагаться атрибуты.

2) Следует перетащить нужные атрибуты слева на право.

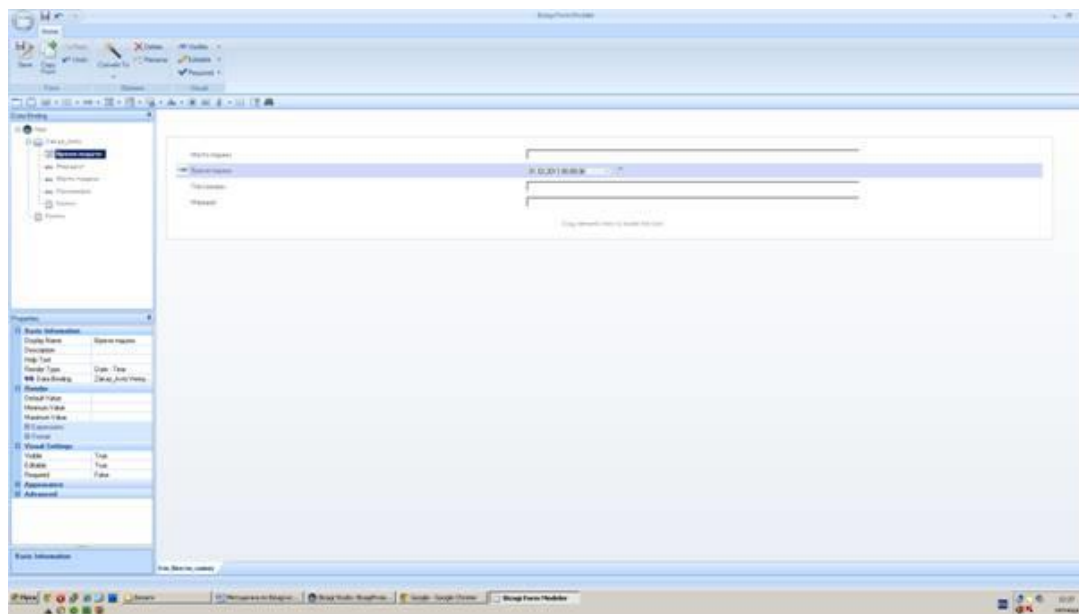



Рисунок 6 - Форма

3) Время подачи нужно уточнить, нужно не просто дата, а **Дата Время**.

Справа на форме, кликнуть по атрибуту Время подачи, найти в самом низу пункт **Convert to** далее **Date-Time** далее **Date-Time**.

Для ввода заявки такой формы вполне достаточно.

4) Необходимо определить форму для каждой задачи. Для ускорения процесса можно нажать на кнопку **Copy From**  и позаимствовать дизайн с уже имеющегося шага, выбрать образец и нажать **Ок** и дизайн скопируется.

Это не очень удобно т. к. если необходимо внести изменения, то придется изменять каждую из форм. Поэтому рекомендуемый способ немного другой:

5) Определить некий повторно используемый компонент. Некое визуальное представление для этой сущности.

Слева в модели данных необходимо раскрыть пункт **Zakaz_Avto** где будут находиться атрибуты, кроме них в самом низу будет

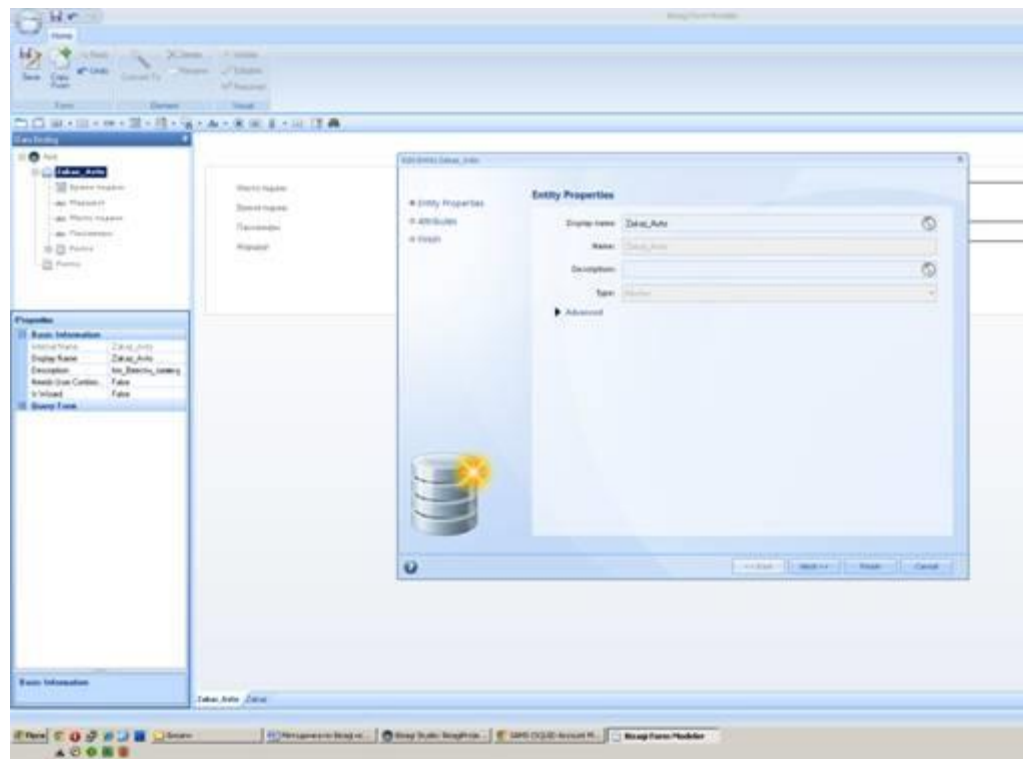


находиться пункт **Forms**. Кликнуть правой кнопкой и далее **New Form**. Уже известным образом перекинуть из модели данных нужные атрибуты. Так же нужно задать имя **Zakaz** (поле **Display Name**) и время.

6) Так же можно сделать поля обязательными для заполнения, кликнуть справа на форме правой кнопкой, далее **Edit Display Form** после произойдет переход в окно редактирования этой формы. Далее правой кнопкой на атрибуте **Required** и там указать **True**. После закрытия появляется форма, ассоциированная с этой сущностью. Теперь её необходимо вставить вместо атрибутов, которые были уже вставлены ранее. Таким образом, форма представляет единое целое. Нужно вернувшись к предыдущей форме.

7) Так же необходимо заявку сделать закрытой для редактирования. Для этого кликнуть правой кнопкой мыши и на форме и выбрать пункт **Editable** и выбрать **False**.

8) А вот решения нет. Т. е. нет атрибута, в котором сохранится решение. Можно выйти, и вернуться к шагу моделирования данных, и добавить там нужный атрибут. Но можно сделать это проще: не выходя из окна редактирования формы, слева на модели данных кликнуть правой кнопкой мыши на пункт **Zakaz_Avto** выбрать **Edit Entity**.



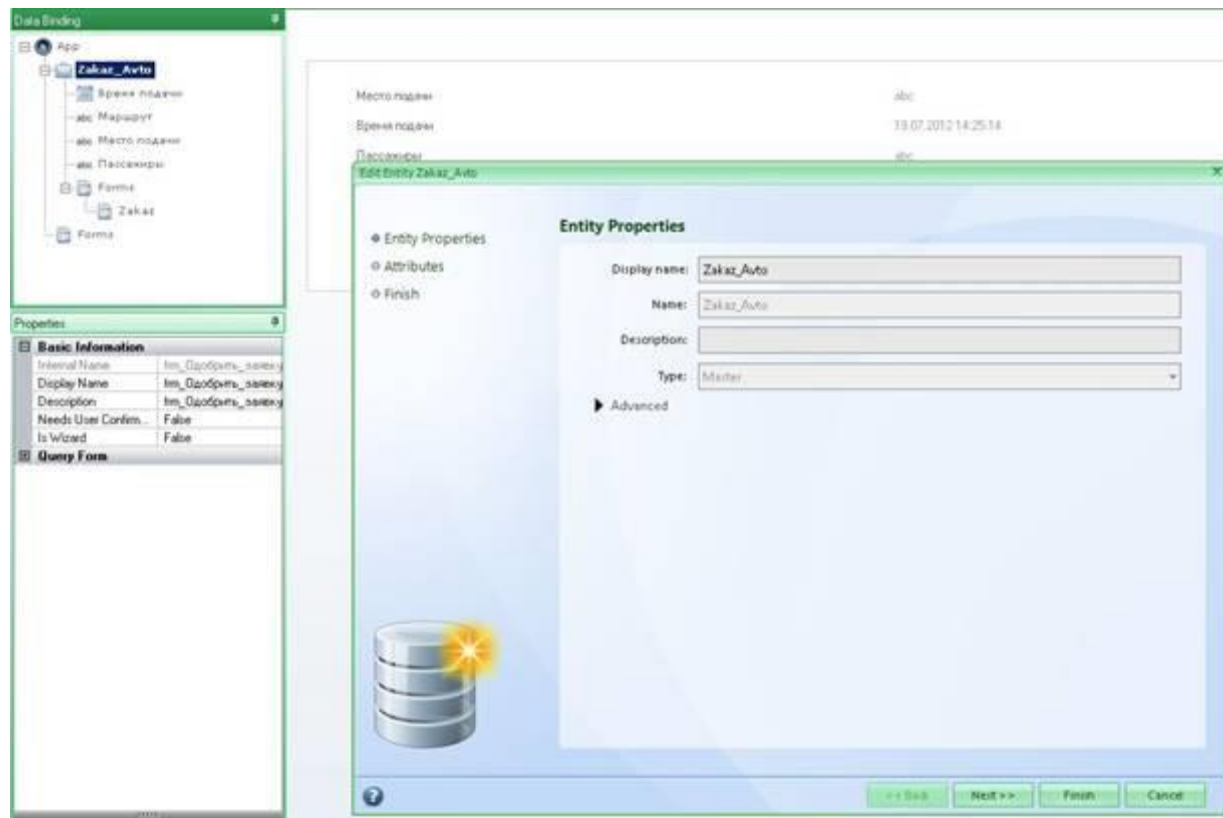





Рисунок 7 – EntityProperties

Далее **Next** и отобразится список атрибутов. Тут *необходимо добавить необходимый атрибут под названием «Одобрено»*. Так же Атрибут **«Принято»** для следующего шага. Тип данных у этих атрибутов должен быть **Boolean**. Готово.



9) Далее добавить атрибут «**Одобрено**» на форму «**Одобрить заявку**». Можно поменять вид атрибута на **Checkbox**. Так же лучше сделать по умолчанию **Одобрено** (для этого нужно в **Properties** найти **Default Value** и указать **True**). Так же можно украсить форму, добавив на неё визуальные элементы, отделить визуально **Заявку** и **Решение**. (Компонент **Adds a group**) .

10) Следующая форма «**Принять заявку**», для неё в качестве образца лучше взять предыдущую форму (кнопка **Copy From**) . Единственное отличие в группе Решение: вместо атрибута **Одобрено - Принято**.

11) Следующей будет форма для задачи Машина не выделена. Здесь в группе решение нужно вместо атрибута *вставить статичный текст, извещающий о том, что машина не выделена*. Для этого используется компонент **Adds a Label** .

12) Следующая форма «**Выполнить рейс**», в качестве образца взята предыдущая форма. Текст изменён на: «Машина выделена. Выполнить рейс».

13) Следующая форма Машина выделена, в качестве образца взята предыдущая форма. Текст изменён на: «Ждите. Машина выделена.».

Практическое занятие № 5 - Редактирование бизнес модели в BizagiModeler

Цель работы: -

5.4 Условия перехода.

Последнее что необходимо сделать для того чтобы нажать кнопку Run и запустить процесс *это определить условия перехода.*



1) Для этого необходимо перейти к 4 разделу разработки «Business Rules». Если вспомнить, то, *были нарисованы стрелки, но не было указано, когда по какой стрелке идти. Наэтомшагеесть 2 пункта, Define Expression и Activity Actions.* Для определения условий перехода нужен 1 пункт.

Загрузив его, появляется схема процесса, где выделены желтым цветом стрелки шлюзов.

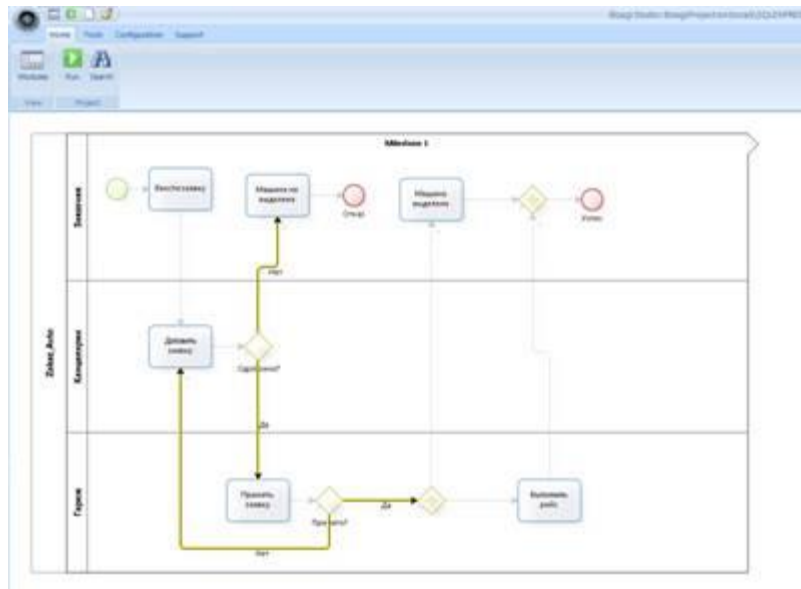


Рисунок 8 - Define Expression

2) Шлюз «Одобрено», стрелка «Да»:

Кликнуть левой кнопкой мыши по стрелке, откроется окно Boolean Expression. В этом окне, слева Модель данных, в которой есть сущность Zakaz_Avto, раскрыв которую можно увидеть её атрибуты. Когда должен сработать переход? Когда значение атрибута Одобрено равно True. Для этой стрелки нужен атрибут Одобрено. *Этот атрибут необходимо перетащить в правую часть, а именно в поле <insert avalue>. В пункте <insert an operator> выбрать «is equal than», а в <insert a value> выбрать True. Finish.*

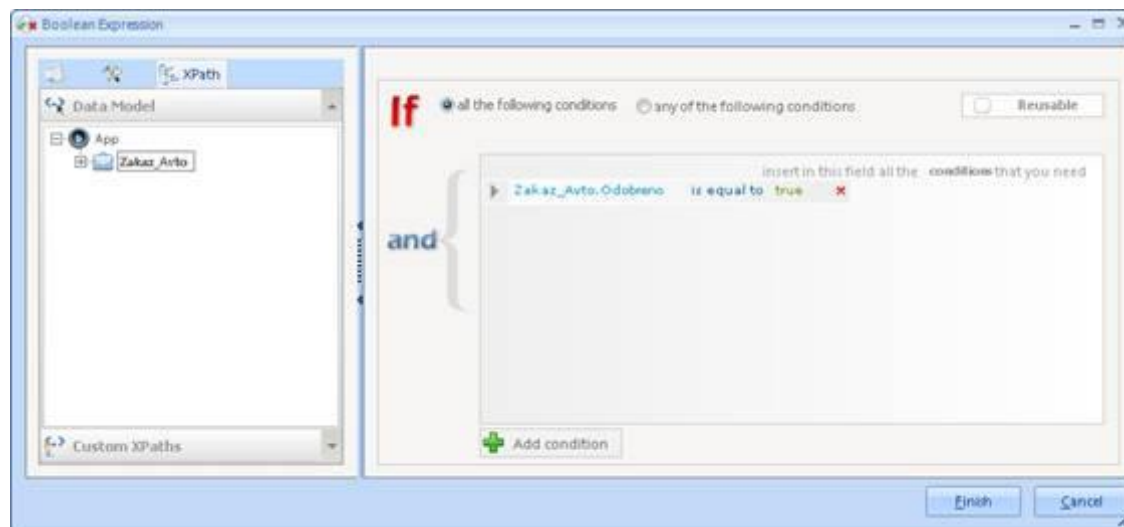


Рисунок 9 - Boolean Expression

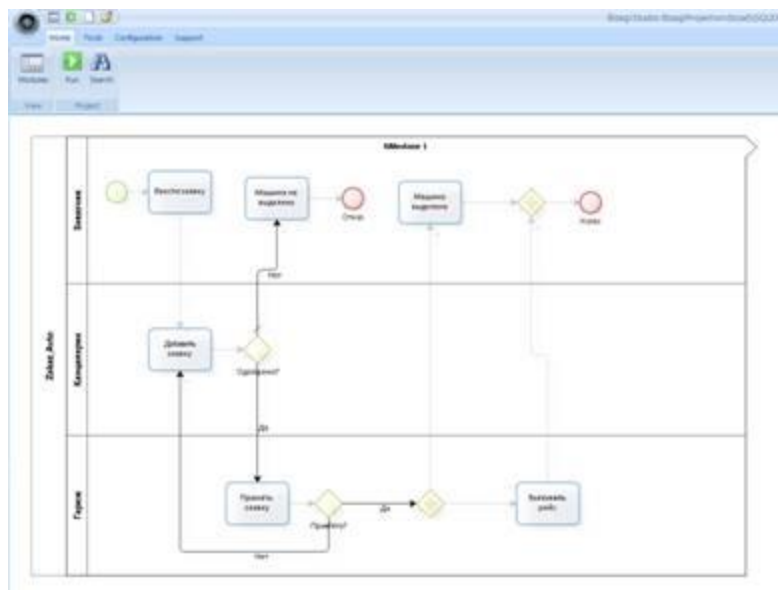
3) Парная стрелка Нет. Для неё конечно можно аналогично указать, когда Одобрено равняется False. Лучше это сделать методом «Иначе». *Для этого нужно закрыть первое окно (Boolean Expression) и произойдёт автоматический переход к ко к окну с названием Expression Selection выбрать и Is Else и нажать на кнопку Ок. Появляется BPMN Defolt flow (зачеркнутая стрелка).*

4) Аналогично со вторым шлюзом, используется атрибут Принято.




Рисунок 10 – IsElse

Парную стрелку методом **Is Else**.




5.5 Работа с порталом Bizagi

1) Теперь можно нажать на кнопку **Run**  и запустить созданный проект. После нажатия, запускается портал **Bizagi**, приложение скомпилируется.

2) После запуска нужно запустить процесс, для этого кликнуть на вкладку Процессы, там есть пункт **Запустить**.

Примечание. Портал не запрашивает имя пользователя и пароль, а так же какой процесс запустить. Это происходит по тому, что существует 1 пользователь и 1 процесс. После регистрации нового пользователя или добавлении процесса такие запросы появятся.

3) Первой появляется форма **Ввести заявку**. Заполнив заявку, нажать на кнопку **Дальше** .

4) Следующей появляется форма «Одобрить заявку». Оставить решение положительным. В форме «Принять заявку» так же оставить положительное решение.

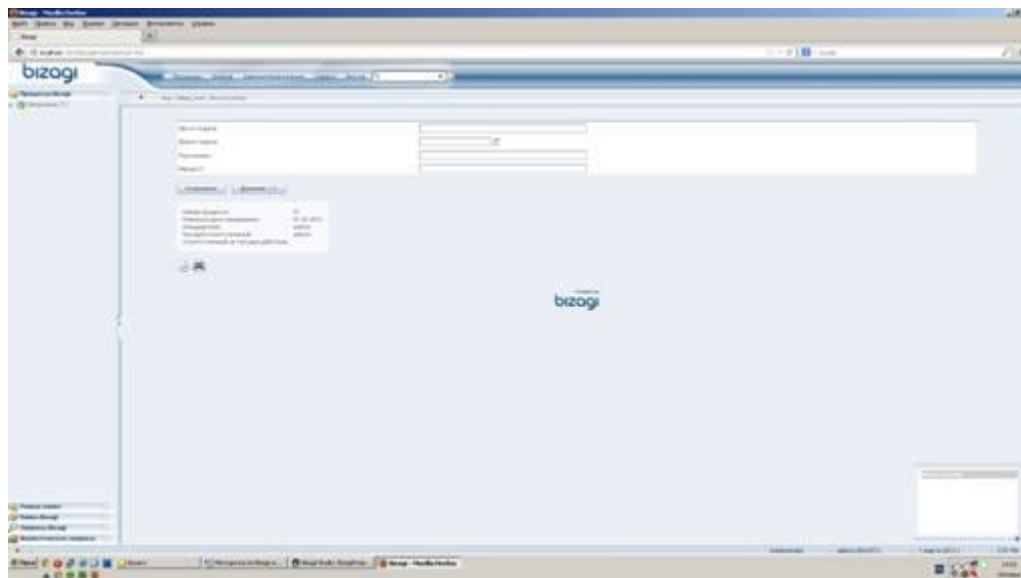


Рисунок 11 – Bizagi портал

5) Далее видны, как и в схеме процесса 2 распараллеленные задачи.

A screenshot of the Bizagi portal displaying a table of running tasks. The table has columns for 'Идентификатор задачи', 'Тип', 'Процесс', 'Состояние', 'Дата создания', 'Срок создания', 'Срок процесса', and 'Принято'. The data shows multiple parallel tasks in progress.

Идентификатор задачи	Тип	Процесс	Состояние	Дата создания	Срок создания	Срок процесса	Принято
2	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да
11	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да
12	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да
13	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да
14	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да
15	AppProcess	Zmag_Ans	- Добавлять заявку	1 марта 2013 г.	- 4 марта 2013 г.	1 марта 2013 г.	Да

Рисунок 12 – Исполняющиеся задачи

б) Можно увидеть эти 2 задачи другим образом. *Кликнуть Процессы >Исполняющиеся.* Можно кликнуть **Просмотр** и увидеть, как процесс исполнялся.

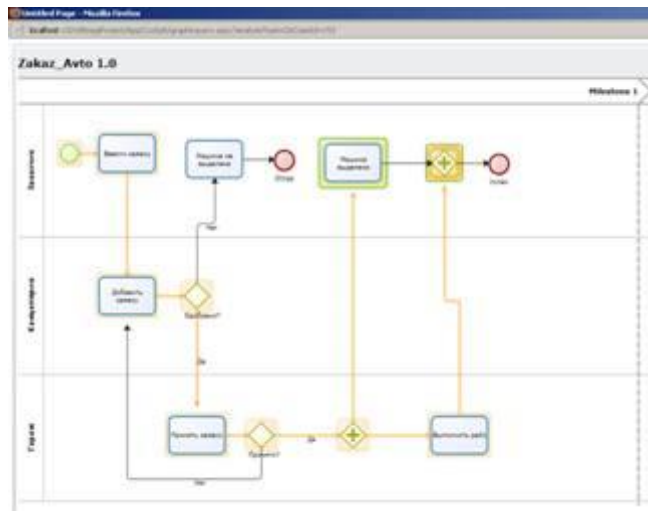


Рисунок 13 - **Просмотр**

5.6 Работа с более сложными структурами данных.

Необходимо дать возможность заказчику указывать, какой тип автомобиля он хочет.

1) Это возможно осуществить, перейдя к шагу 2, моделирование данных. Для таких целей нужно добавить ещё одну таблицу и подключить её к существующей, но легче работать в одной, и **Bizagi** позволяет это делать.

2) Открыть **лист атрибутов (Edit Attribute List)**. Добавить атрибут определяющий тип машины. А вот *тип* будет ссылка на справочник. Т. е. на новую сущность.

3) Выбрать тип **Parameter**, потому что в данном случае будет **справочник**. И **New Entity** потому что такого справочника пока что нет.

4) После нажатия на **New Entity** появляется определение этой сущности. Имя этой сущности **TipMash**. Так же нужно указать атрибуты этой сущности (1 атрибут, **Тип машины** тип **String**). Далее **Finish**. На рисунке видно, что начинает выстраиваться **Er – диаграмма**. Видно, что заявка имеет *ссылочный атрибут* **Тип машины**, который ссылается на сущность и там есть один атрибут **Тип машины**.

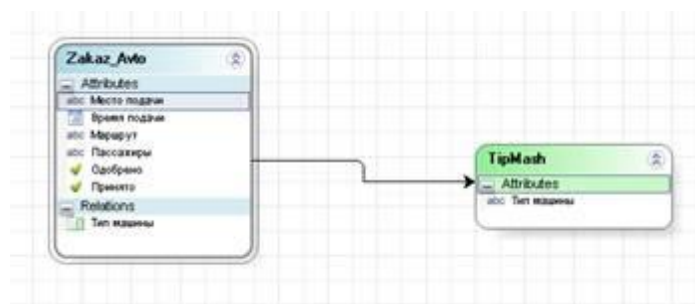

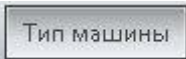


Рисунок 14 - **Тип машины**

5) Сразу, прямо отсюда можно наполнить справочник, какими ни будь значениями. Кликнув правой кнопкой мыши выбирать **Values**, нажать на кнопку **Тип машины** в низу окна, появляется 1 значение.

Примечание. Есть кнопка добавить  , и нет кнопки удалить. Т. е из справочника данные принципиально никогда не удаляются. Единственное, что можно сделать, пометить как не используемые (**Disabled**). Это сделано что бы гарантировать ссылочную целостность, чтобы удаление какого-то значения не исказило данные уже имеющиеся.

6) Теперь необходимо «прицепить» эти данные к форме. На вводе заявки появится возможность выбора тип автомобиля. *Открыть форму* «**Ввести заявку**», далее **Edit Display Form**, и перенести туда атрибут «**Тип машины**». Атрибут был добавлен в общий блок, соответственно необходимость в редактировании остальных форм отпадает.



Рисунок 15 – Форма Тип машины

Можно удостовериться и открыть остальные формы.

После этих действий необходимо запустить процесс и проверить внесенные изменения.



Рисунок 16 – Заполненная форма Тип машины

Выбор из справочника очень распространенный сценарий. Второй не менее распространенный сценарий это заявка, в которой есть многострочная часть. Примером будет список затрат. Т. е. когда водитель сделал рейс, вернулся. Необходимо чтобы у него была возможность составить авансовый отчет. Где он может указать, сколько он потратил на бензин, замену колеса и т. п. Как это реализовать?

7) Аналогично предыдущему, начать с уже существующей таблицы. *Перейти к моделированию данных. Edit Attribute List, и создать атрибут под названием затраты.* Но тип у этого атрибута будет необычный, **Collection**. Здесь связь будет не N к одному, а N к N , т. е. одной заявки соответствует N записей этих затрат. Далее все аналогично. **Master**, потому что это не справочник а фактические данные, **New Entity**. Создать сущность под названием **Zatrati**.

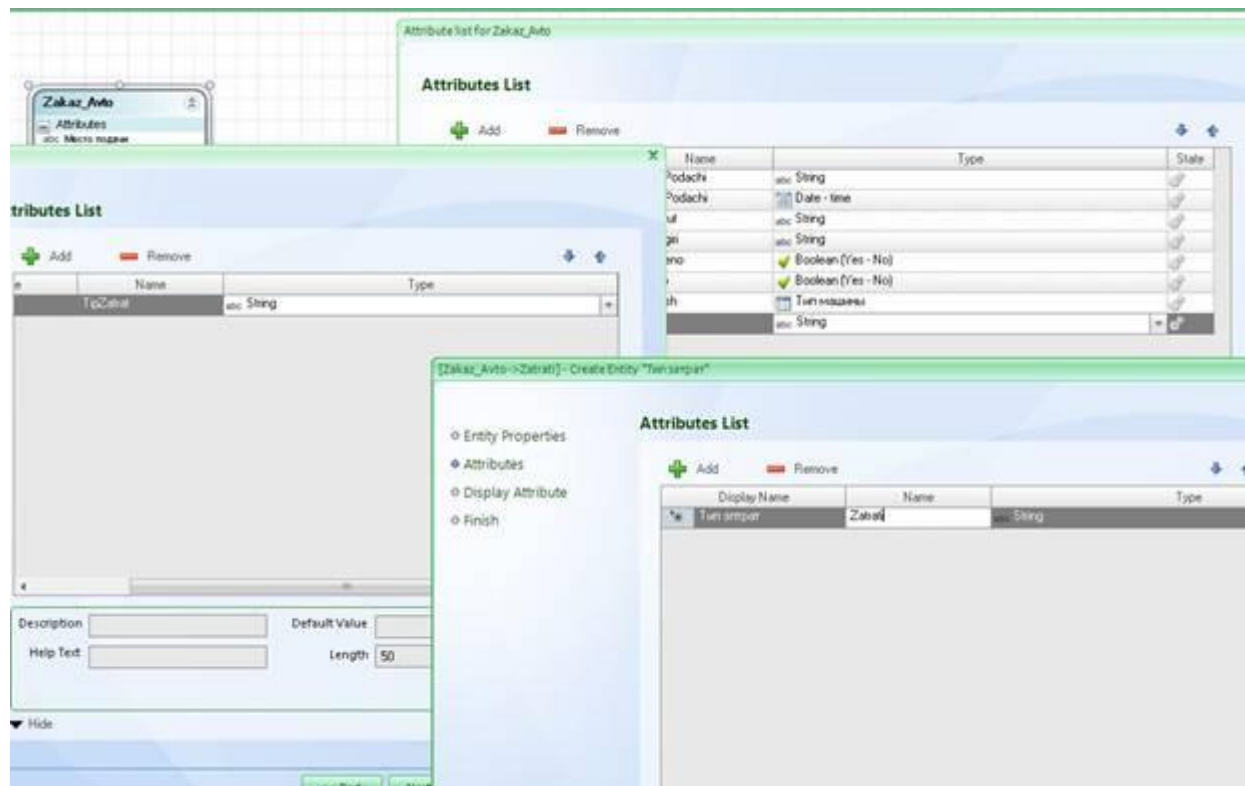


Рисунок 17 – ModelData

Атрибутами будут:

1. Типы затрат (имя **TipiZatrat**, значения в поле **Name** должны отличаться). Это будет справочник, Так же как и в предыдущей сущности. **Entity**> **Parameter**> **New Entity**. Имя атрибута справочника Типа затрат (**TpZatrat**).
2. Документ (**Doc**), основание. Тип **String**.
3. Сумма (**Suma**). Тип **Currency**.

Готово. Заявка ссылается на множество записей затрат, те в свою очередь на справочник типов затрат.

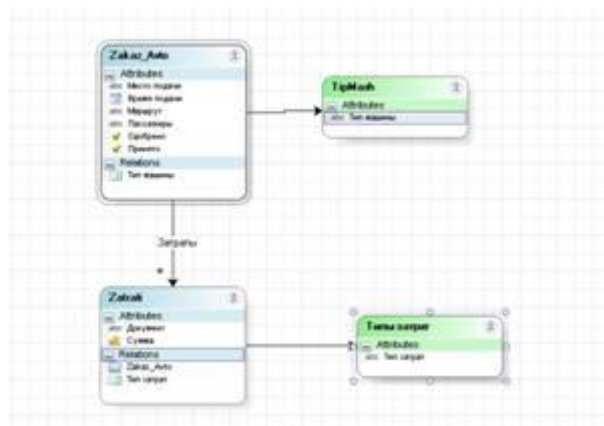


Рисунок 18 - Затраты

[Получить полный текст](#)



8) Вставить эти затраты в форму а именно в форму задачи «**Выполнить рейс**». Открыть форму, и вставить атрибут затрат на форму. Там появится таблица затрат, где можно добавлять и удалять строки.

9) Далее необходимо определить внешний вид этой таблицы.



Рисунок 19 – Форма Затраты


Для этого **Display Form** > **New Form**. Имя **Zatrati**. Перекинуть туда атрибуты. Сохранить, закрыть. Далее необходимо разрешить удаление. Выделить табличку и в свойстве **Allow Delete** ставим **True**. Для улучшения внешнего вида сделать следующее:


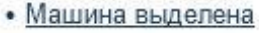
10) В свойствах есть вкладка **Labels**  и вместо **Add** Затраты вписать следующее **Ещё затраты**, а вместо **Delete** написать **Удалить**. Ещё можно указать итогов (Свойства **Advanced** > **Grid Totalizers** > далее кнопку **Add**  далее поменять в **Display Text** текст **SUM** на **Итого**).

11) Необходимо проверить результаты, нажать кнопку **Run**.

Заполнить все формы, переходя от шага к шагу. Выбирать «**Выполнить рейс**». Нажать «**Ещё затраты**» [Ещё Затраты](#) .

Но как видно справочник Тип затрат пустой. Можно вернуться к разработке и там его наполнить, так же портал даёт возможность администрирования, позволяя наполнять справочники нужными данными, не выходя из портала.

Выбирать Администрирование > Тип затрат. Нажать на кнопку Добавить Тип затрат [Добавить Типы затрат](#) . Ввести необходимые данные и нажать на кнопку сохранить  (были созданы следующие типы затрат ГСМ, Сервис, Прочее).

Далее перейти в исполняющиеся процессы, найти там запущенный процесс нажать на Выполнить рейс  [Выполнить рейс](#) . Так же создать список затрат.  [Машина выделена](#) .

Заполнить так как показано на рисунке 20.

Заказчик: admin
 Номер заказа: 351
 Дата Заказа: 20 июля 2012 г.
 Место подачи: Ул. Софьи Перовской
 Время подачи: 23 июля 2012 г. 13.42
 Пассажиры: 3
 Маршрут: Ул. Рабочее
 Тип машины: Представительская

Удалить	Тип затрат	Документ	Сумма
<input type="checkbox"/>	ГСМ	чек	123,00
<input type="checkbox"/>	Сервис	квитанция	1 231,00
			Итого: 1 354,00р.

Еще Затраты
 Линейный отчет:

Рисунок 20 – Заполненная форма Затраты

5.7 Автоматизация.

До этого момента всё происходило вручную.

Для дальнейшего усовершенствования процесса *нужно добавить 3 атрибута: Заказчик, Номер заказа и Дата заказа.*

1) Далее в моделирование данных и создать эти атрибуты.




У атрибута **Заказчик** тип будет не строка, а ссылка на системную сущность.

Тип > Entity > System > WFUSER. Таким образом Будут указаны данные зарегистрированного пользователя.

Номер заказа – тип **String**. Дата заказа **Date Time**.

2) Перейти ко второму шагу разработки, формам, открыть общую форму (**Edit Display Form**). Найти сущность **Заказчик**, раскрыть и перенести атрибут **fullName** форму. Поменять имя атрибута на **Заказчик (Display Name)**. Так же нужно перенести атрибуты **Номер заказа** и **Дата заказа**. Лучше запретить редактирование этих атрибутов. **Сохранить**.

Примечание. Если не сделать атрибут **Заказчик** не редактируемым то будет ошибка, и не удастся сохранить форму.

3) Осталось определить автоматические действия. Перейти в раздел **Business Rules**, нажать на **Activity Action** . Прямо на входе в первой задаче вычислить эти действия. Кликнуть на задачу **Ввести заявку**, появится окно **Bizagi Dialog**, выбираем **On Enter**  **On Enter**, нажать на  в нижнем левом углу окна, там выбрать **Expression**. Появится новое окно, ввести имя в поле **Name (Compute)**. Далее правой кнопкой на стрелке и выбрать **Add Expression**. Далее **Ок**.

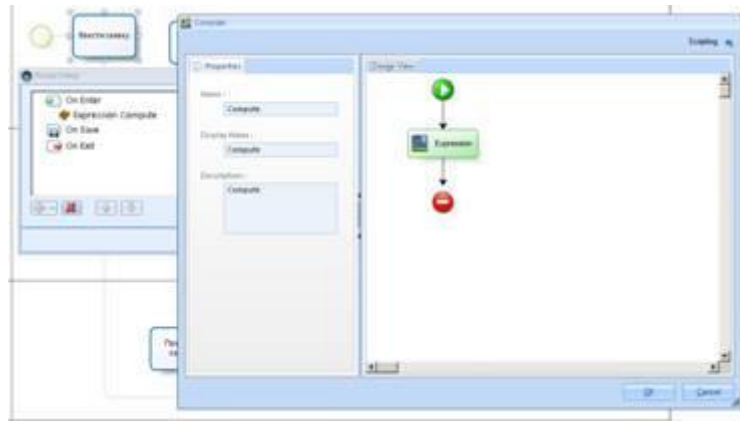


Рисунок 21 - **Expression**

Два раза кликнуть по появившейся иконке появится окно редактирования **Edit Expression**. Для кодирования в **Bizagi** используется **Visual J Sharp** к которому добавлены некие способы адресации к полям.

4) Обращение к полям происходит следующим образом:

Ввести скобку: <

Появится сущность, выбирать её, далее ввести точку, и выбирать нужный атрибут.

Потребуется атрибуты **Заказчик, Номер заказ и Дата заказа**. Также потребуются функция **Me**.

Примечание. Есть описание на сайте Bizagi (ссылка на описание этой функции http://wiki. /en/index. php?title=Advanced_Function:_Me).

Код должен выглядеть следующим образом:

<Zakaz_Avto. Zakazchik> = Me. Case. Creator. Id;

<Zakaz_Avto. NomerZakaza> = Me. Case. CaseNumber;

<Zakaz_Avto. DateZakaza> = Me. Case. CreationDate;

При синтаксических ошибках появляются сообщения. Если всё **«чисто»** можно запускать.

5) Запустить процесс. Видно, что появились созданные атрибуты Заказчик, Номер заказа и Дата заказа. Но отличие в том что эти атрибуты заполняются сами. Т. е. есть 1 пользователь (**admon**), и все данные берутся оттуда.



The screenshot shows a web form with the following fields and values:

Заказчик:	admon
Номер заказа:	351
Дата заказа:	10 мая 2010 г.
Место подачи:	<input type="text"/>
Время подачи:	<input type="text"/>
Паспорт:	<input type="text"/>
Маршрут:	<input type="text"/>
Тип машины:	<input type="text"/>

At the bottom of the form, there are two buttons: "Сохранить" (Save) and "Далее >>" (Next).

Рисунок 22 – Автоматически заполненная форма Ввести заявку

5.8 Подготовка печатных документов.





Создание авансового отчета.


Водитель ввел некие расходы, лучше сделать их в виде некоторой справки.


1) Снова нужно перейти к разделу моделирование данных. Добавить специального вида атрибут.

Авансовый отчет (**AОтчет**), *тип специальный для автоматических документов* **Template (More Types> Template)**.

2) Этот атрибут должен присутствовать на форме «**Выполнить рейс**». Нужно перейти в **Define Forms** и перекинуть туда этот атрибут.

3) Далее нужно перейти в **Business Rules> Activity Action** . Далее кликнуть на задаче **Выполнить рейс**. Отчет будет строится после действия **Сохранить**  **On Save**. Далее кликнуть на , и выбрать **Letter> New**. Создать *новый шаблон* . С тем же самым названием. Далее **Display Name** - Авансовый отчет. Далее указать созданный специальный атрибут (**Авансовый отчет**).

4) Далее, окно формы, ввести заголовок **Авансовый отчет**. А вместо номера указать из атрибута, нажатием на **Xpath Field** , там показаны атрибуты, выбрать **Номер заказа**.

5) И вставить **динамическую таблицу (Dynamic Grid)** . *Двойной клик на таблице* здесь нужно задать, откуда брать коллекцию записей для формирования этой таблицы. Для этого кликнуть на **Select** в поле **Fact Name**, и найти там атрибут **Zatrati**, далее **Ок**.

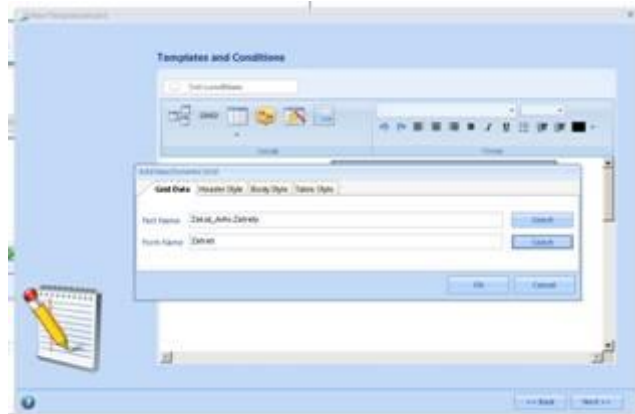


Рисунок 23 – TemplateWizard

6) Далее указать форму через которую отображаться отчет, **Form Name**, так же **Select** выбирать **Zatrati** и нажать **Ok**. Далее **Ok**.




Далее, контрольный список покажет, что все было сделано правильно.

7) Необходимо запустить процесс и проверить результаты. Дойдя до задачи выполнить рейс, ввести затраты. Прежде чем смотреть отчет нужно нажать кнопку **Сохранить**. Это достаточно примитивный документ. Но легок в реализации.

Тип затрат	Документ	Сумма
ГСМ	чек	111,00р
сервис	квитанция	333,00р

Рисунок 24 Авансовый отчет

5.9 Мониторинг.

Самый примитивный анализ во вкладке **Исполняющиеся**. Здесь представлены все исполняющиеся процессы. Так же имеется возможность посмотреть на динамику процесса, увидеть все переходы. Для этого нужно нажать на кнопке просмотр . Здесь можно запустить имитацию, которая покажет как происходил переход задач в данном процессе. Это можно сделать нажав на кнопку . Так же в этом окне присутствует кнопка , нажатие которой показываются задачи на которых остановился процесс.

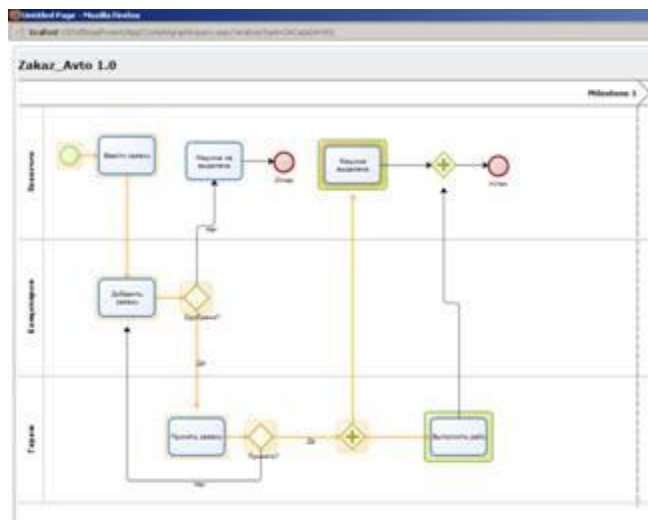


Рисунок 25- Просмотр

Если необходимо посмотреть на картину целом, то не обойтись без раздела **Анализ**, который включает 5 пунктов.

Примечание. Мониторинг это экземпляры процессов которые ещё исполняются а Аналитика – экземпляры процессов, которые уже завершены.

Вкладка анализ состоит из:

1. Мониторинг процессов.

Первая вкладка называется **Анализ загрузки**.

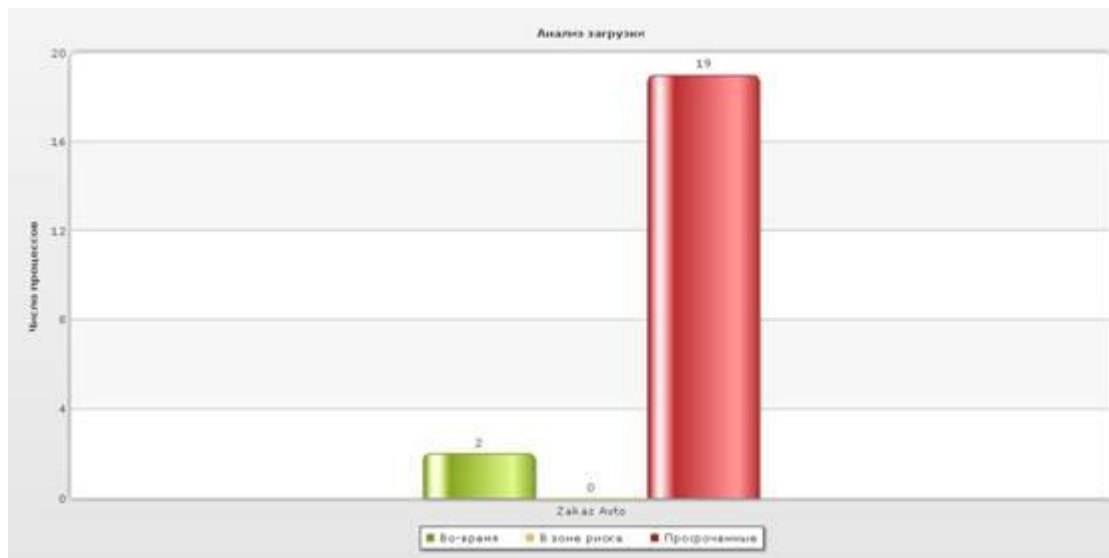
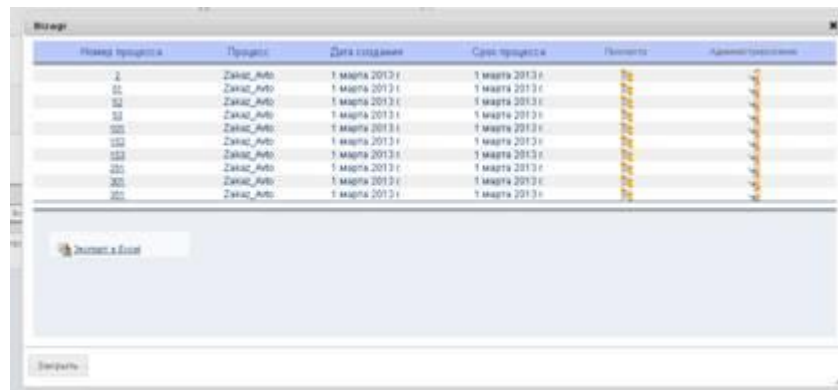


Рисунок 26 - Анализ загрузки

График показывает число активных процессов, имеющих запас по времени (зеленый столбец), находящихся в зоне риска (срок которых истекает сегодня, желтый столбец) и просроченных (красный столбец).



Имя процесса	Продукт	Дата создания	Срок процесса	Статус	Администратор
2	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Зеленый	Администратор
21	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
22	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
23	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
24	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
25	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
26	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
27	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
28	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
29	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор
30	Заказ_Авто	1 марта 2013 г.	1 марта 2013 г.	Желтый	Администратор

Рисунок 27 – Подробности Анализа загрузки

При клике на один из столбцов будут показаны подробности (данные всех процессов отнесенных в эту группу).

Вторая закладка. Работа не завершена.

Состоит из двух графиков:

Состояние процесса: График показывает процент процессов с запасом по времени, в зоне риска (срок которых истекает сегодня) и просроченных. Подробности по клику на диаграмме.

Состояние процесса

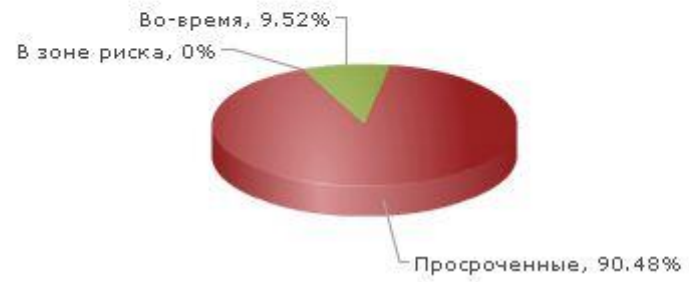


Рисунок 28 - Состояние процесса

Процессы с истекающим сроком: График показывает когда открытые процессы окажутся просрочены. Подробности по клику на диаграмме.

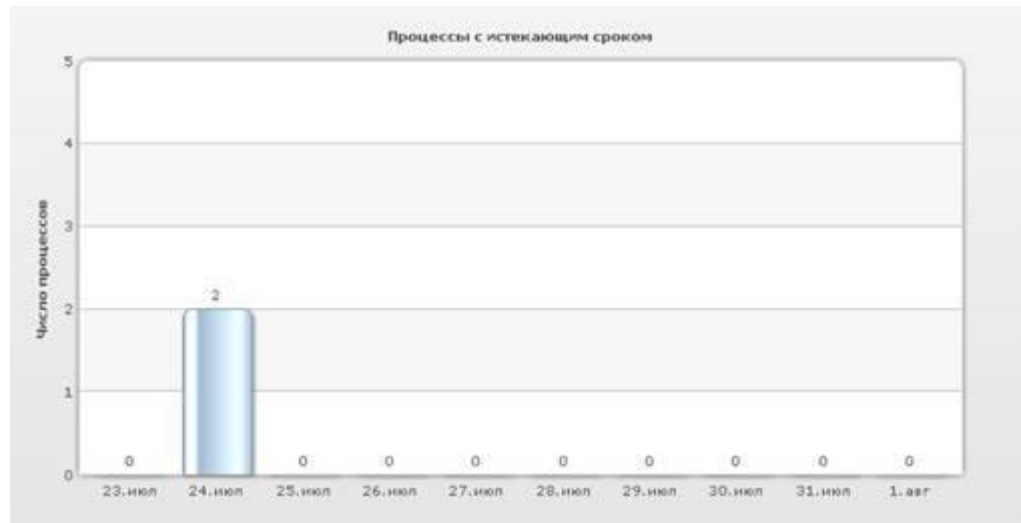
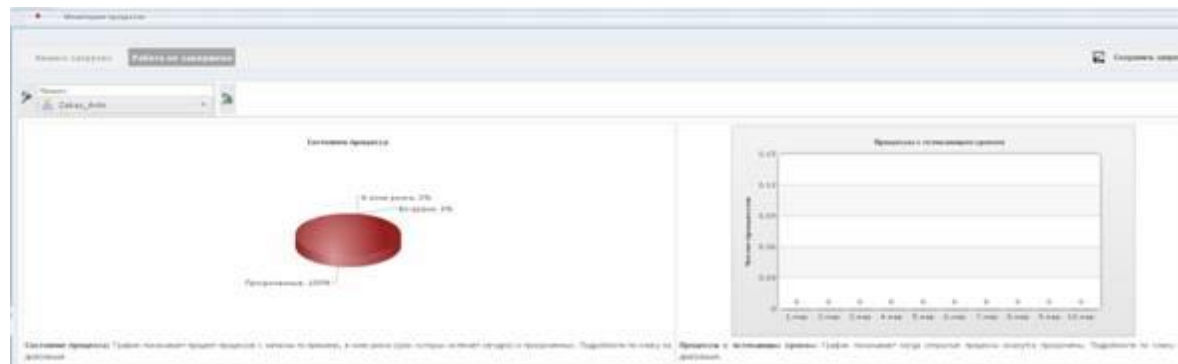


Рисунок 29 - Процессы с истекающим сроком



2. Мониторинг задач.

Аналогичный график. Но уже не в разрезе процессов, а задач.

Первая закладка. Так же состоит из двух графиков.

Состояние задачи: График показывает процент заданий с запасом по времени, в зоне риска (срок которых истекает сегодня) и просроченных. Подробности по клику на диаграмме.

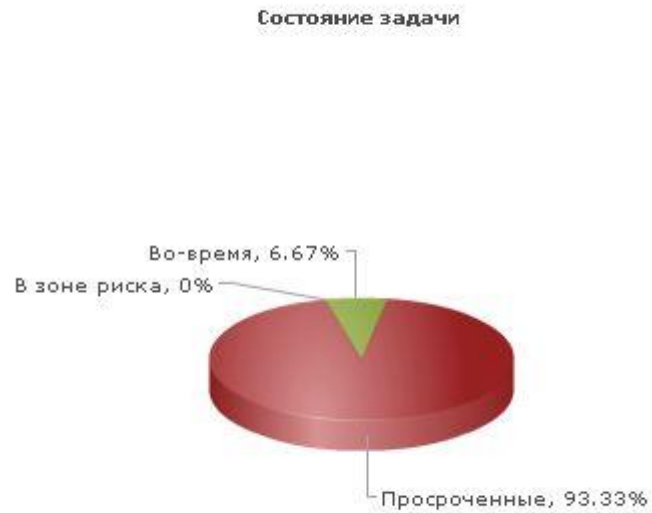


Рисунок 30 - Состояние задачи

Задачи с истекающим сроком: График показывает когда открытые задания окажутся просрочены. Подробности по клику на диаграмме.

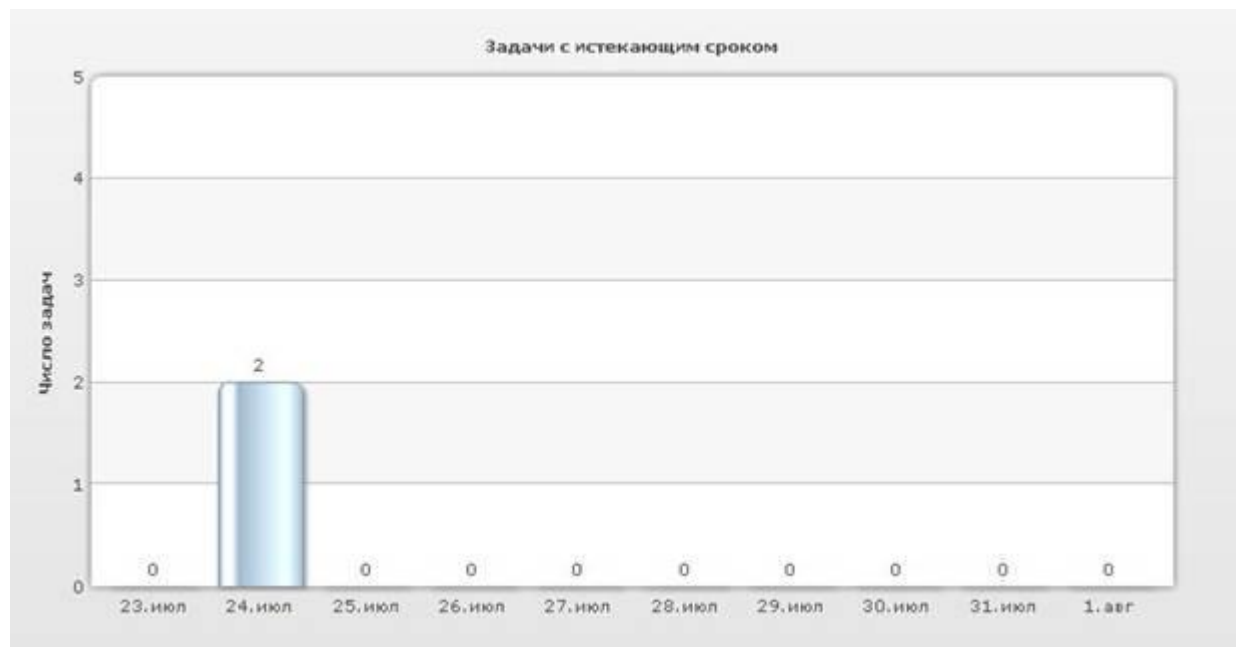
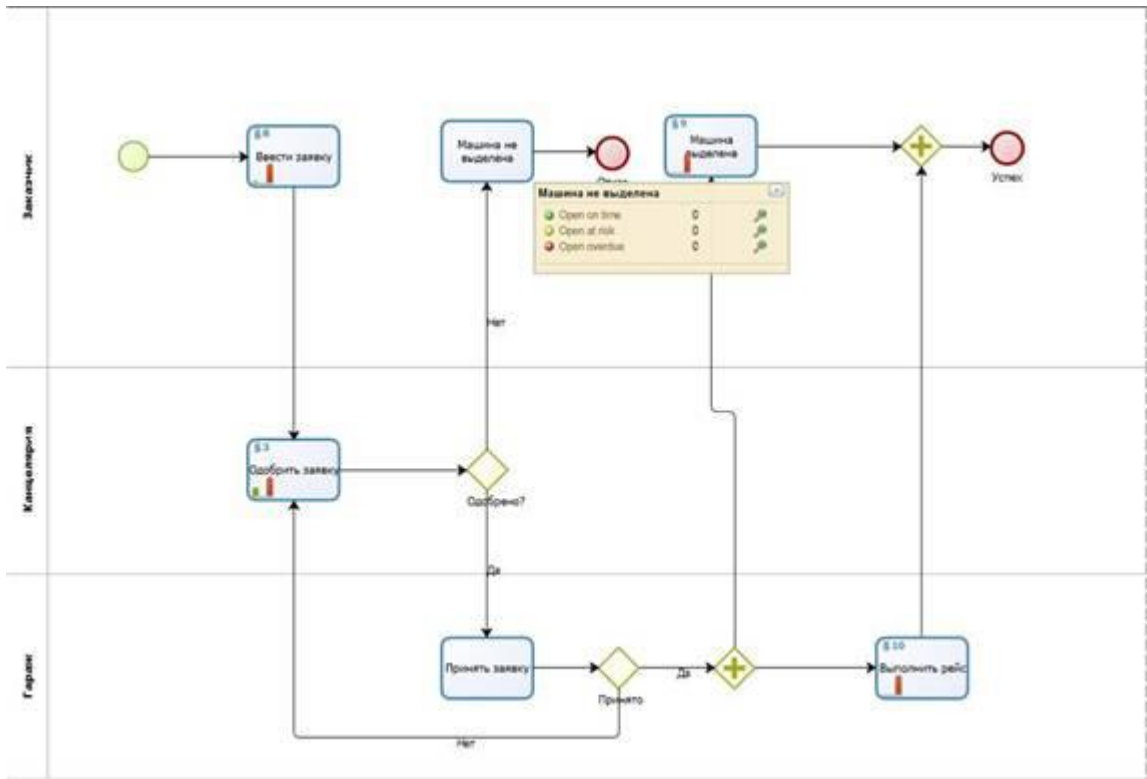


Рисунок 31 - Задачи с истекающим сроком



Вторая вкладка Анализ загрузки. Показывает узкие места в процессе.



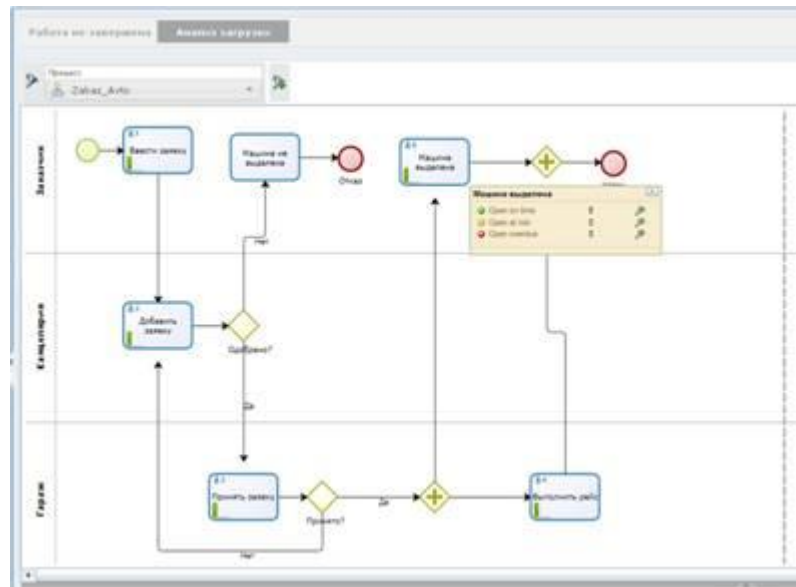


Рисунок 32 - Анализ загрузки

3. Аналитика процессов

Продолжительность. Это статистика. Можно выбрать период для расчета.

Состоит из двух графиков:

Продолжительность процесса: Средняя продолжительность по сравнению с ожидаемой.

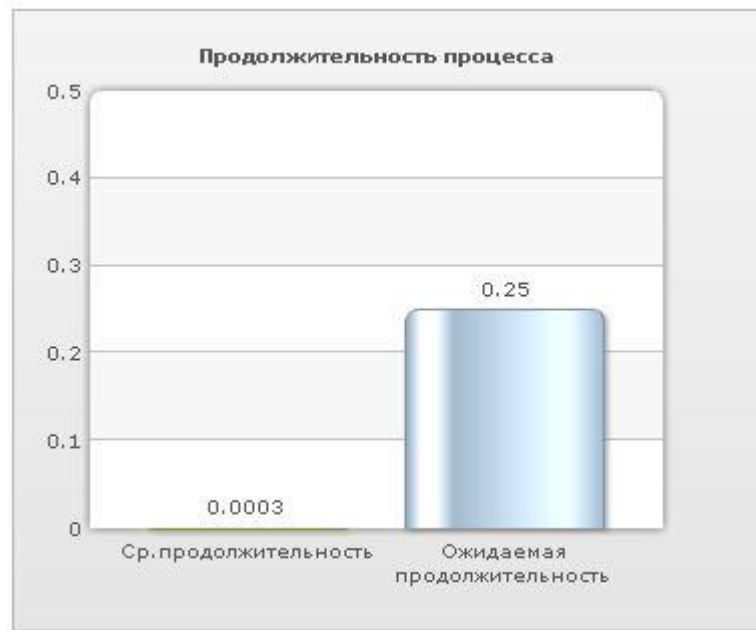


Рисунок 32 - Продолжительность процесса

Состояние процесса: Процессы, закрытые в срок и с превышением. Подробности по клику на диаграмме.

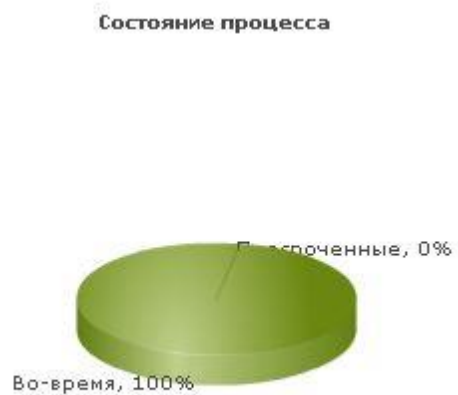


Рисунок 33 - Состояние процесса

Вторая вкладка. Гистограмма продолжительности: График показывает сколько дней заняло выполнение завершившихся процессов. Вертикальный пунктир отделяет завершённые вовремя процессы от просроченных. График можно построить только если процесс выполнялся более 1 дня.

Третья вкладка. Процессное задание. Показывает сколько за какой-то промежуток времени процессов появилось и сколько завершилось.

Состоит из двух графиков:

Процессное задание: Процессы запущенные, завершённые и прерванные в выбранном интервале времени. Подробности по клику на диаграмме.

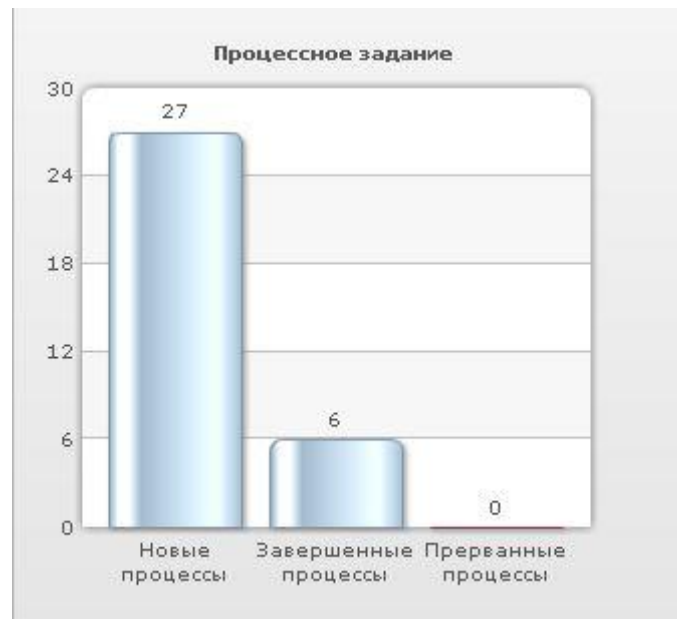


Рисунок 34 - Процессное задание

Тренд: Тренд числа запусков процесса в выбранном временном интервале.

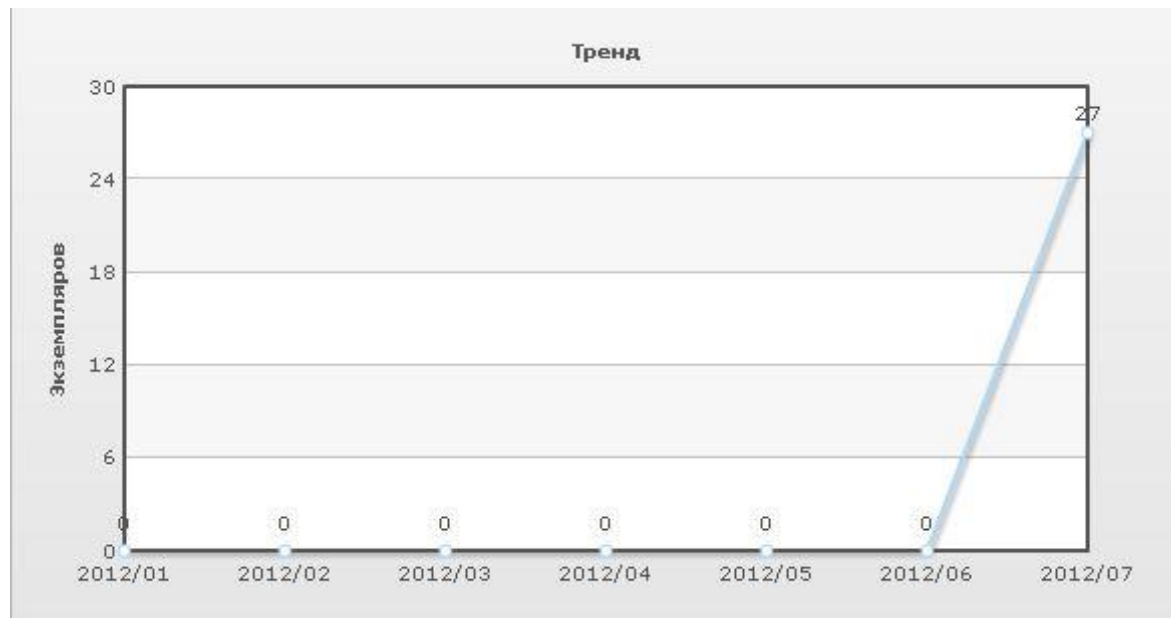


Рисунок 35

Четвертая закладка.

Часто запускаемые процессы: Процессы, чаще других запускаящиеся в выбранном временном интервале. Подробности по клику на диаграмме. Так как создан все 1 процесс, данный график обладает малой информативностью.

Пятая закладка. Повторяющиеся маршруты. Показывает наиболее популярные сценарии развития процесса.

Первым по популярности является маршрут, конечным состоянием которого является Успех.

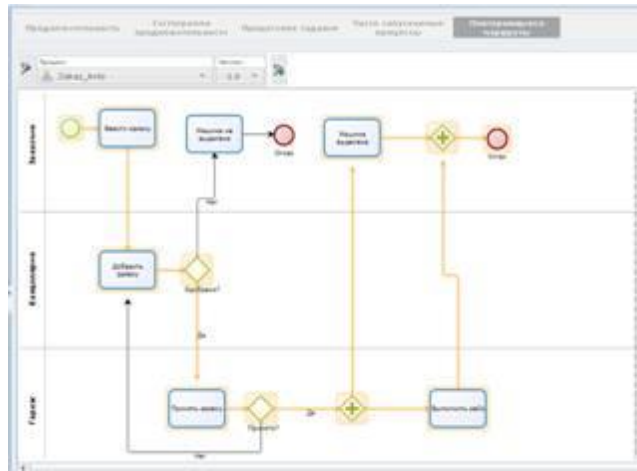


Рисунок 36 - Повторяющиеся маршруты

4. **Аналитика задач.** Здесь можно увидеть, как задачи укладываются в норматив или не укладываются.

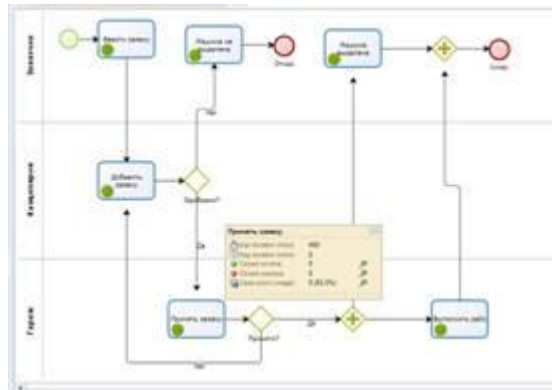
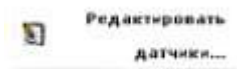
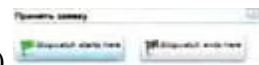


Рисунок 37 - Аналитика задач

5. **Аналитика контрольных точек.** С помощью этого инструмента создать некий контрольный отрезок, кликнув на

«**Редактировать датчик**»  . Далее «**Новая контрольная точка**». Далее нужно задать начальную и конечную задачу выбрав соответствующий пункт (нужно *навести курсор мыши на задачу*)  , сохранить это, в каком то виде (в «**свойствах секундомера**» ввести название в поля Наименование и Экранное наименование). Так же нужно задать **SLA** (1 час).

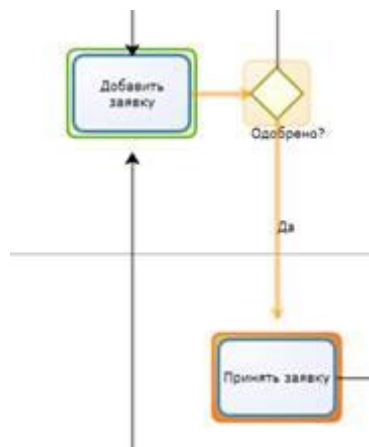


Рисунок 38 - Аналитика контрольных точек

Практические занятия № 6

Построение компьютерной сети по созданной бизнес модели

Целью работы является изучение технологии построения компьютерных сетей

2. Задачи

закрепление и расширение теоретических знаний студентов, приобретение навыков в работы в PacketTracer

Задание лабораторной работы

Компания арендовала 3 помещения в бизнес центре. В этих помещениях есть только голые стены и розетки. Вы друг основателя фирмы и по совместительству сетевой и системный администратор. Вас попросили разработать схему сети.

В сети должна быть реализована возможность связываться с любым из трёх помещений в компании, но при этом каждое помещение (отдел) должны быть изолированы.

Также в третьем помещении необходимо создать беспроводную точку доступа. Эта точка должна иметь пароль junior17, должны автоматически выдаваться первые 20 адресов, SSID должен быть скрыт.

Во втором отделе стоит не настроенный web сервер. Это тоже необходимо исправить. От Вас требуется реализовать в каждом помещении возможность получать доступ к серверу по url имени.

В первом отделе 4 рабочих места, во втором — 2 рабочих места и сервер, третье помещение нужно для отдыха персонала (10 рабочих мест, в том числе 4 беспроводных).

К сетевому оборудованию вам необходимо предоставить безопасный удаленный доступ (SSH). Обеспечить защиту портов доступа на коммутаторах (не более 2 адресов на интерфейсе, адреса должны быть динамически сохранены в текущей конфигурации, при попытке подключения устройства с адресом, нарушающим политику, на консоль должно быть выведено уведомление, порт должен быть отключен).

Ход работы:

Инструкция по выполнению лабораторной работы в PacketTracer

1. Добавление оборудования.

Открыть PacketTracer и создать на рабочем поле:

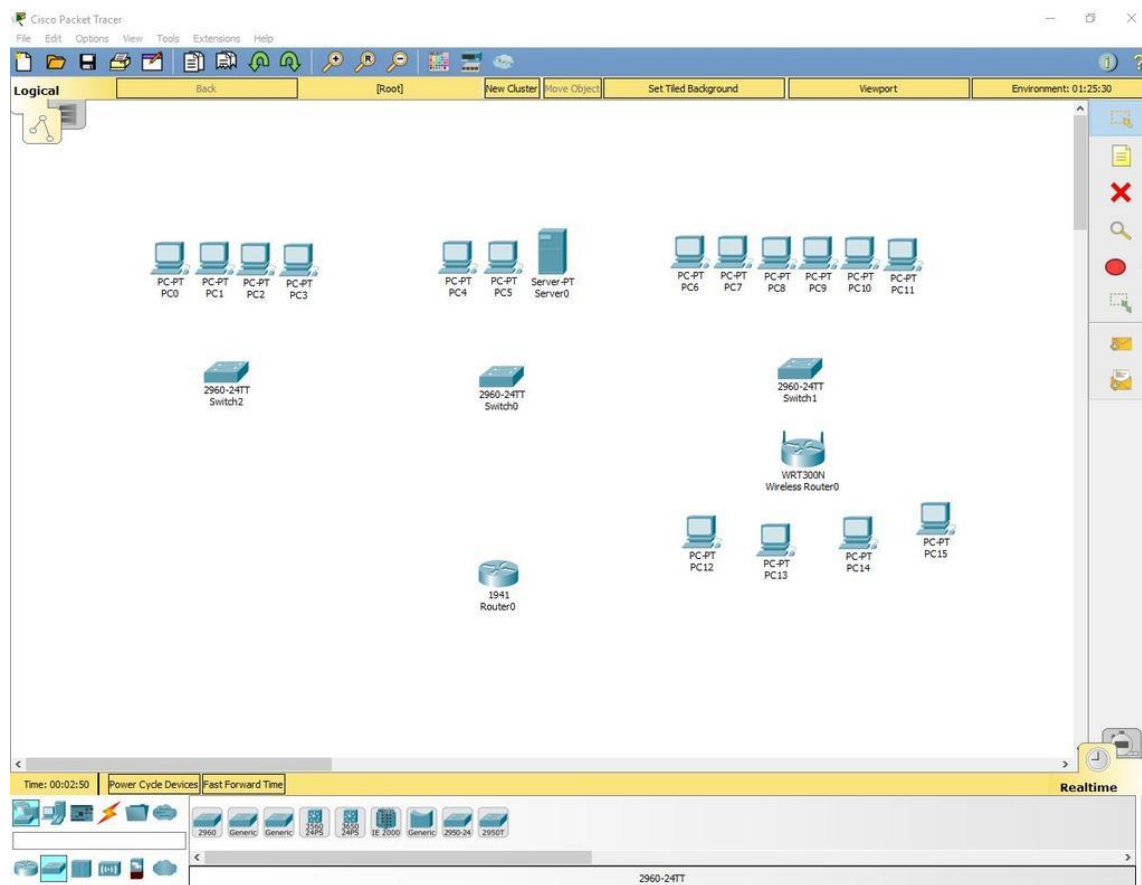
a. 16 компьютеров

b. Сервер

c. 3 коммутатора Cisco 2960

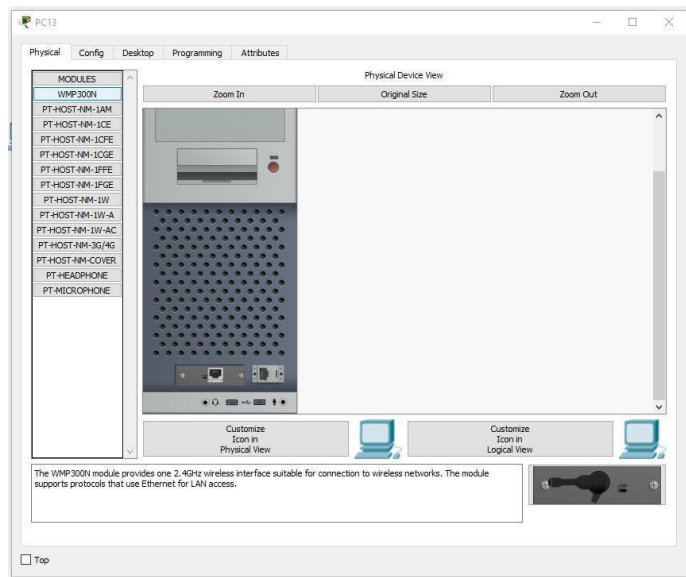
d. Маршрутизатор Cisco 1941

e. Роутер Cisco WRT300N



2. Установка Wi-Fi модуля в ПК.

У четырёх компьютеров в третьем отделе заменить LAN разъём на Wi-Fi антенну. Для этого открываем устройство, выключаем его, вынимаем старый модуль, меняем его на Wi-Fi (WMP300N) антенну. Включаем компьютер.



3. Настройка ПК первого и второго отдела.

Каждому компьютеру в первом и втором отделе, а также серверу присвоим значения по формуле: $N0.0.0.n$, где N – номер отдела, а n – номер устройства (например, $10.0.0.2$ – второй компьютер на первом этаже). Сервер, так как он третье устройство на втором этаже будет иметь адрес $20.0.0.3$.

Маску подсети выставим на $255.255.255.0$.

DefaultGateway выставим $N0.0.0.254$.

DNS Server выставляем на $20.0.0.3$.

4. Настройка третьего отдела.

Выставим IP по формуле $30.0.0.10n$, где n – номер ПК.

Продолжим настройку ПК. Первый IP – $30.0.0.101$, а последний – $30.0.0.110$

5. Настройка роутера.

Выставим настройки:

IP - 30.0.0.253

Маска - 255.255.255.0

Start IP Address - 30.0.0.1

Maximum number of Users – 20

Static DNS 1 - 20.0.0.3

Network Name - Cisco2107

SSID Broadcast – Disabled

Security Mode - WPA2-Personal

Passphrase - junior17

Настройка беспроводных ПК. Задаём имя сети Cisco2107 и WPA2-Personal пароль – junior17

6. Подключаем кабели и соединяем отделы.

Соединяем ПК витой парой.

Во всех коммутаторах подключаем кабели к FastEthernet по часовой стрелке. В маршрутизаторе подключимся к гигабитному разъёму, предварительно его включив.

Настраиваем VLAN на всех коммутаторах. Для этого открываем коммутатор в первом отделе. Переходим в интерфейс командной строки и вводим команды:

```
Switch>en  
Switch#conf t
```

```
Switch(config)#vlan 10
Switch(config-vlan)#name Office1
Switch(config-vlan)#end
```

Рассмотрим все команды.

1. En – enable. Расширенный доступ к конфигурации
2. Conf t – Configurationterminal. Открывает терминал настройки
3. Vlan 10 – создаёт виртуальную сеть с индексом 10
4. Name Office1 – задаётся имя VLAN. Имя – Office1.
5. End – завершения настройки.

Открываем коммутатор во втором отделе и прописываем следующие команды:

```
Switch>en
Switch#conf t
Switch(config)#vlan 10
Switch(config-vlan)#name Office1
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Office2
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name Office3
Switch(config-vlan)#exit
Switch(config)#end
```

Открываем коммутатор в третьем отделе и прописываем следующие команды:

```
Switch>en
Switch#conf t
Switch(config)#vlan 30
```



```
Switch(config-vlan)#name Office3  
Switch(config-vlan)#end
```

Выставляем на первом коммутаторе VLAN 10 на все порты, к которым есть подключение (Fa0/1-Fa0/5).

На втором коммутаторе нужно выставить порт, к которому подключен коммутатор из первого отдела VLAN – 10, из третьего VLAN – 30, а 2 ПК и сервер второго отдела VLAN – 20. То есть Fa0/1 – VLAN 10, Fa0/2- Fa0/4 – VLAN 20, Fa0/5 – VLAN 30. Fa0/6, соединяющий коммутатор и маршрутизатор выставляем в Trunk режим.

На третьем коммутаторе нужно выставить на все порты VLAN 30 (Fa0/1-Fa0/8).
Затем, производим настройку маршрутизатора для работы с VLAN.

Также, переходим во вкладку CLI и прописывает там команды:

```
Router>en  
Router#conf t  
Router(config)#int gig 0/0.10  
Router(config-subif)#encapsulation dot1Q 10  
Router(config-subif)#ip address 10.0.0.254 255.255.255.0  
Router(config-subif)#exit  
Router(config)#int gig 0/0.20  
Router(config-subif)#encapsulation dot1Q 20  
Router(config-subif)#ip address 20.0.0.254 255.255.255.0  
Router(config-subif)#exit  
Router(config)#int gig 0/0.30  
Router(config-subif)#encapsulation dot1Q 30  
Router(config-subif)#ip address 30.0.0.254 255.255.255.0  
Router(config-subif)#end
```

Теперь разберём команды:

1. `int gig 0/0.10`. Команда подключает виртуальный интерфейс для работы с разными VLAN. Цифра после точки – номер VLAN.
2. `Encapsulation dot1Q 10`. Команда настройки VLAN в sub. Номер после `dot1Q` – номер VLAN.

3. `ipaddress 10.0.0.254 255.255.255.0`. IP адрес выхода пакетов информации.

Теперь протестируем сеть командой `ping`.

Возьмём любой компьютер в каждом отделе и пропиnguем все отделы (в третьем отделе проверим и проводную сеть и беспроводную).

Добавляем административный VLAN (40 — Management).

7. Настройка сервера.

Включаем DNS.

Name — `www.cisco.com`.

Address — `20.0.0.3`.

Проверим возможность выхода на сайт из любого отдела. Вводим URL имя в браузере и нажимаем Go.

8. Настроим SSH.

Для этого заходим в маршрутизатор и пишем команды:

```
Router>en
Router#clock set 10:10:00 13 Oct 2017
Router#conf t
Router(config)#ip domain name ssh.dom
Router(config)#crypto key generate rsa
Router(config)#service password-encryption
Router(config)#username Valery privilege 15 password 8 junior17
```

```
Router(config)#aaa new-model
Router(config)#line vty 0 4
Router(config-line)#transport input ssh
Router(config-line)#logging synchronous
Router(config-line)#exec-timeout 60 0
Router(config-line)#exit
Router(config)#exit
Router#copy running-config startup-config
```

Разберём каждую команду:

1. clockset 10:10:00 13 Oct 2017. Устанавливаем точное время для генерации ключа.
2. ipdomainnamesh.dom. Указываем имя домена (необходимо для генерации ключа).
3. cryptokeygeneratorsa. Генерируем RSA ключ (необходимо будет выбрать размер ключа).
4. servicepassword-encryption. Активируем шифрование паролей в конфигурационном файле.
5. username Valery privilege 15 password 8 junior17. Заводим пользователя с именем Valery, паролем junior17 и уровнем привилегий 15.
6. aaanew-model. Активируем протокол AAA (до активации AAA в системе обязательно должен быть заведен хотя бы один пользователь).
7. linevtu 0 4. Входим в режим конфигурирования терминальных линий с 0 по 4.
8. transportinputssh. Указываем средой доступа через сеть по умолчанию SSH.
9. loggingsynchronous. Активируем автоматическое поднятие строки после ответа системы на проделанные изменения.
10. exec-timeout 60 0. Указываем время таймаута до автоматического закрытия SSH сессии в 60 минут.
11. copy running-config startup-config. Сохраняем конфигурационный файл в энергонезависимую память. (Здесьвыведетсястрока «Destination filename [startup-config]?») Вводим «startup-config»).

9. Настроим защиту портив на каждом коммутаторе.

Для этого открываем коммутатор и пишем команды:

```
Switch>en
Switch#conf t
Switch(config)#interface range fastEthernet 0/X-Y
```

```
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport port-security
Switch(config-if-range)#switchport port-security maximum K
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#switchport port-security violation shutdown
Switch(config-if-range)#end
```

Разберём каждую команду:

1. Interface range fastEthernet 0/X-Y. Выбор диапазона интерфейсов (X – первый нужный порт, Y – последний).

ВНИМАНИЕ! Выбирайте порты которые НЕ активны в подключениях!

2. switchport mode access. Переводим порт в access режим.

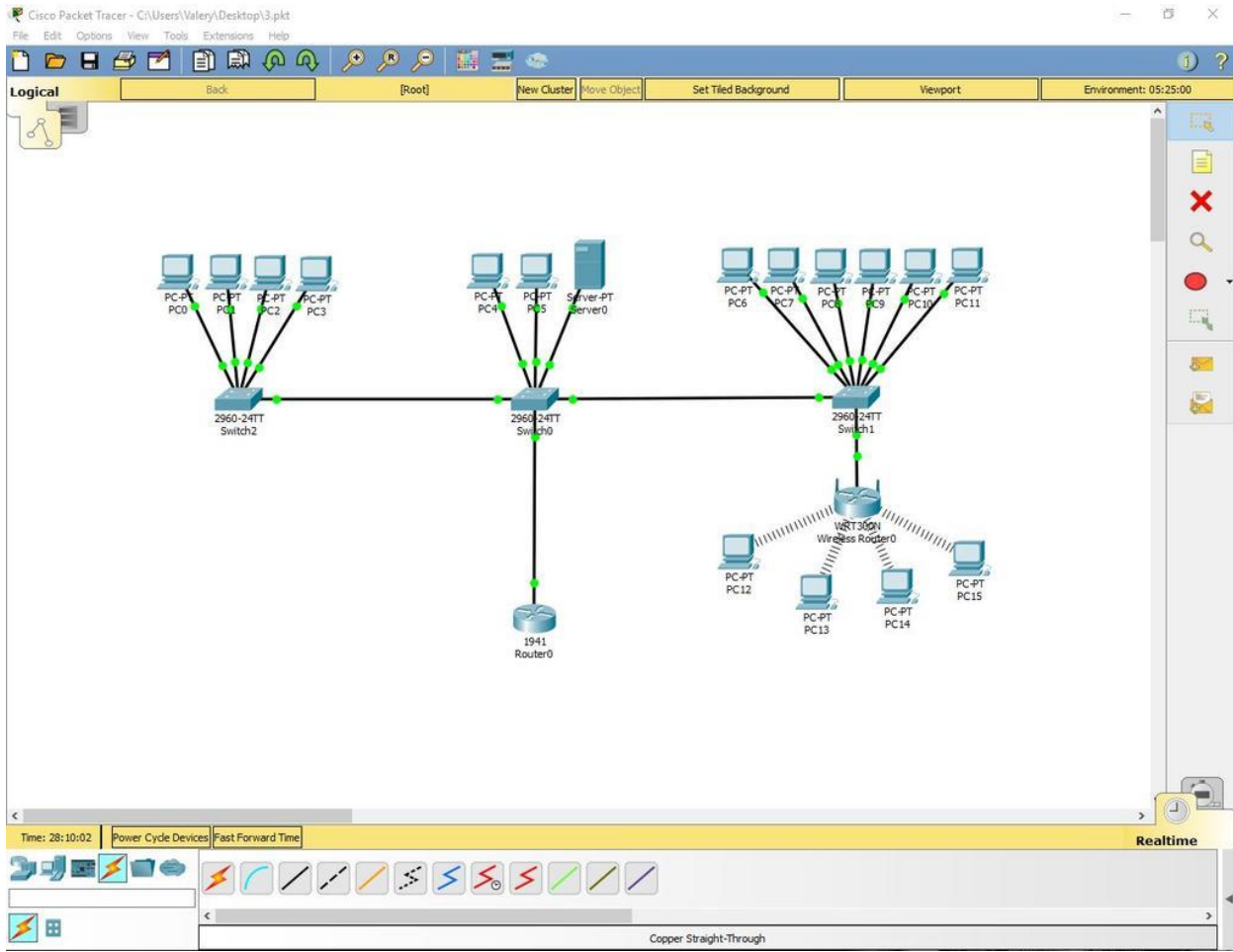
3. switchport port-security. Включаем защиту портов.

4. switchport port-security maximum K. Ограничиваем число MAC-адресов на интерфейсе (K – число портов).

5. switchport port-security mac-address sticky. Выбираем способ изучения MAC-адресов коммутатором (есть статический (mac-address) и динамический (sticky)).

6. switchport port-security violation shutdown. Задаем тип реагирования на превышение числа разрешенных MAC-адресов (бывают protect – после переполнения все пакеты, отправленные с других MAC-адресов отбрасываются, restrict – то же самое, но с уведомлением в syslog или по SNMP, shutdown – порт выключается до автоматического или ручного его поднятия, также отправляются уведомления).

В итоге работа выполнена так:



2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

2.1 Основные источники

Электронные издания (электронные ресурсы)

1. Менеджмент: бизнес-информатика : учебник и практикум для СПО / Е.П. Зараменский. – Москва : Издательство Юрайт, 2019. – 407 с. (серия профессиональное образование)
2. 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, 2016. — 289 с

2.2. Дополнительные источники

1. Моделирование бизнес-процессов: лабораторный практикум /М.Ю. Арзуманян, М.А. Деревянко, СПбГУТ. – СПб.,2014 -48 с.

2.3 Интернет ресурсы:

1. <http://www.intuit.ru>