

bpmntraining.ru

Лабораторная работа по курсу
«BPMN103 – Исполняемый BPMN»

Часть 1: Оркестровка

Оглавление

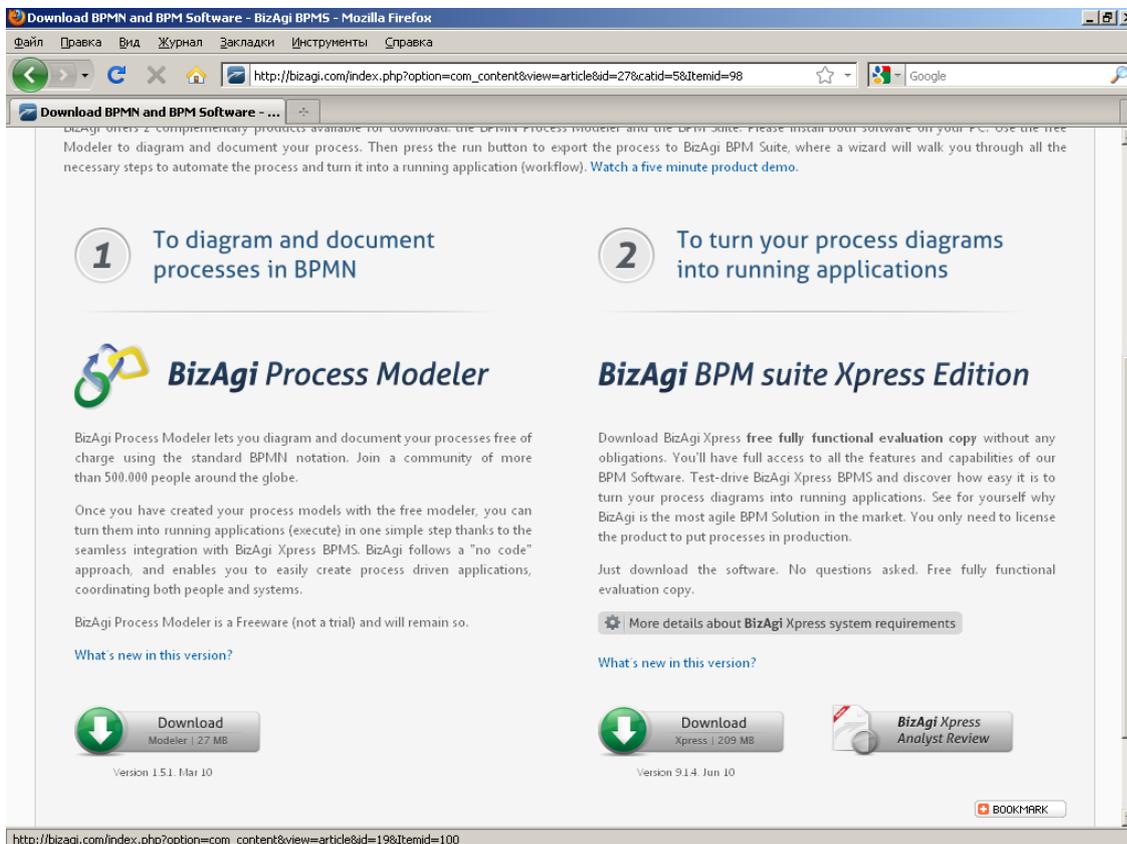
1.	Инсталляция	3
2.	Создание проекта	7
3.	Схема процесса	10
4.	Схема данных	12
5.	Формы	16
6.	Добавление атрибутов на лету	22
7.	Правила переходов	27
8.	Исполнение	30
9.	Справочники	34

1. Инсталляция

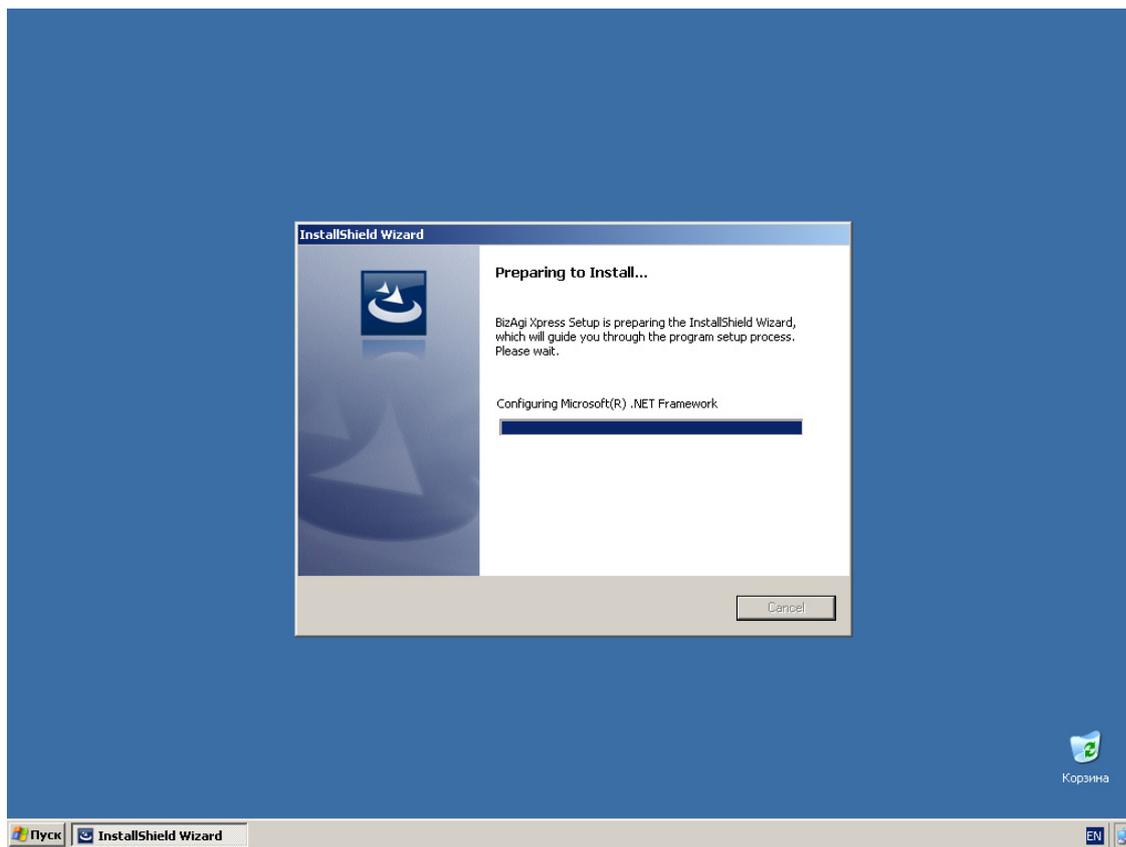
Программное обеспечение BizAgі Xpress свободно скачивается с сайта www.bizagi.com:



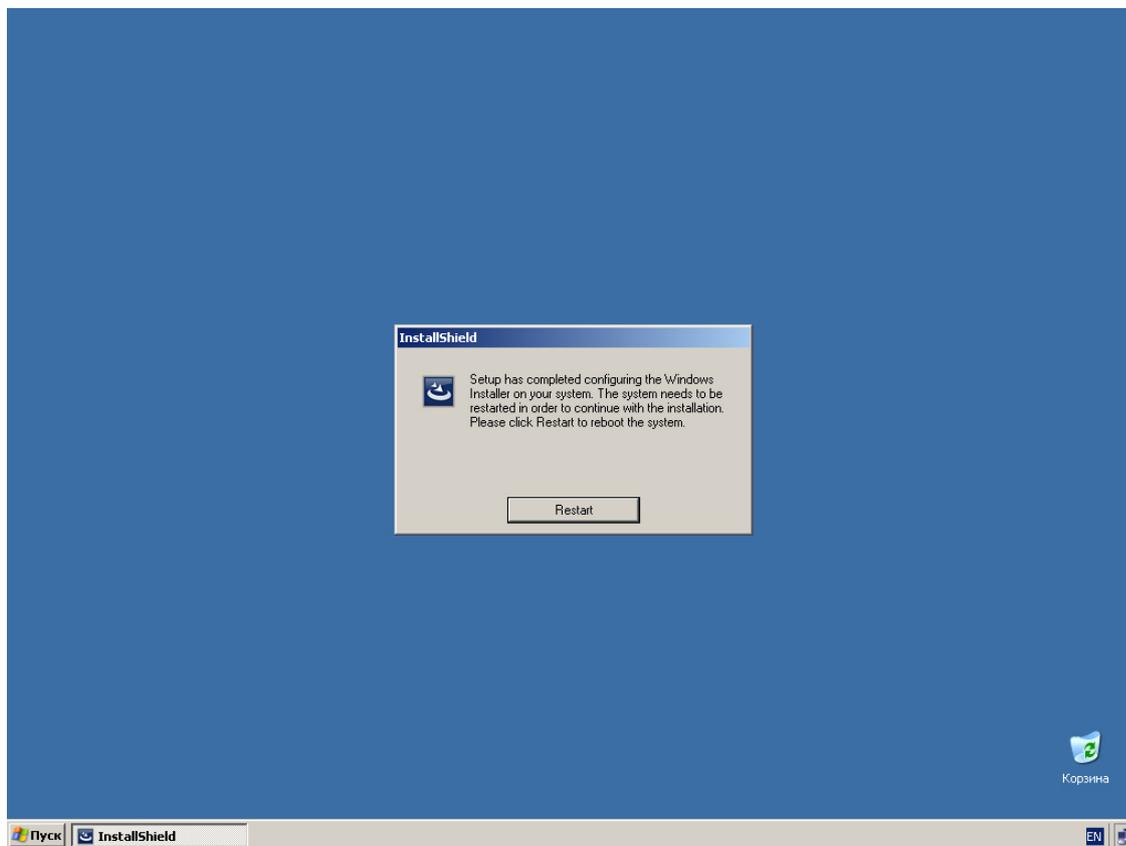
Щелкните по кнопке Download на главной, а на следующей странице щелкните по кнопке Download под заголовком BPM BPM suite Xpress Edition:

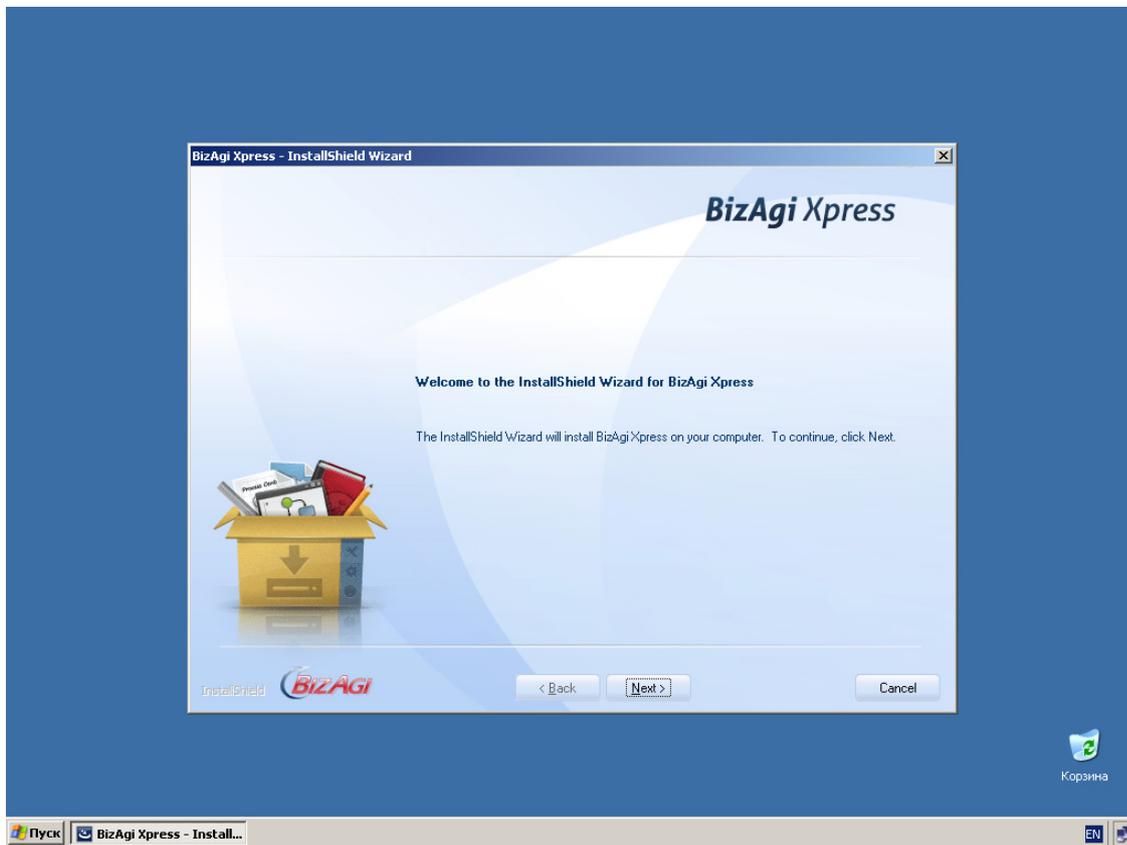


В процессе инсталляции BizAgi Xpress может установить на ваш компьютер программное обеспечение Microsoft .NET.

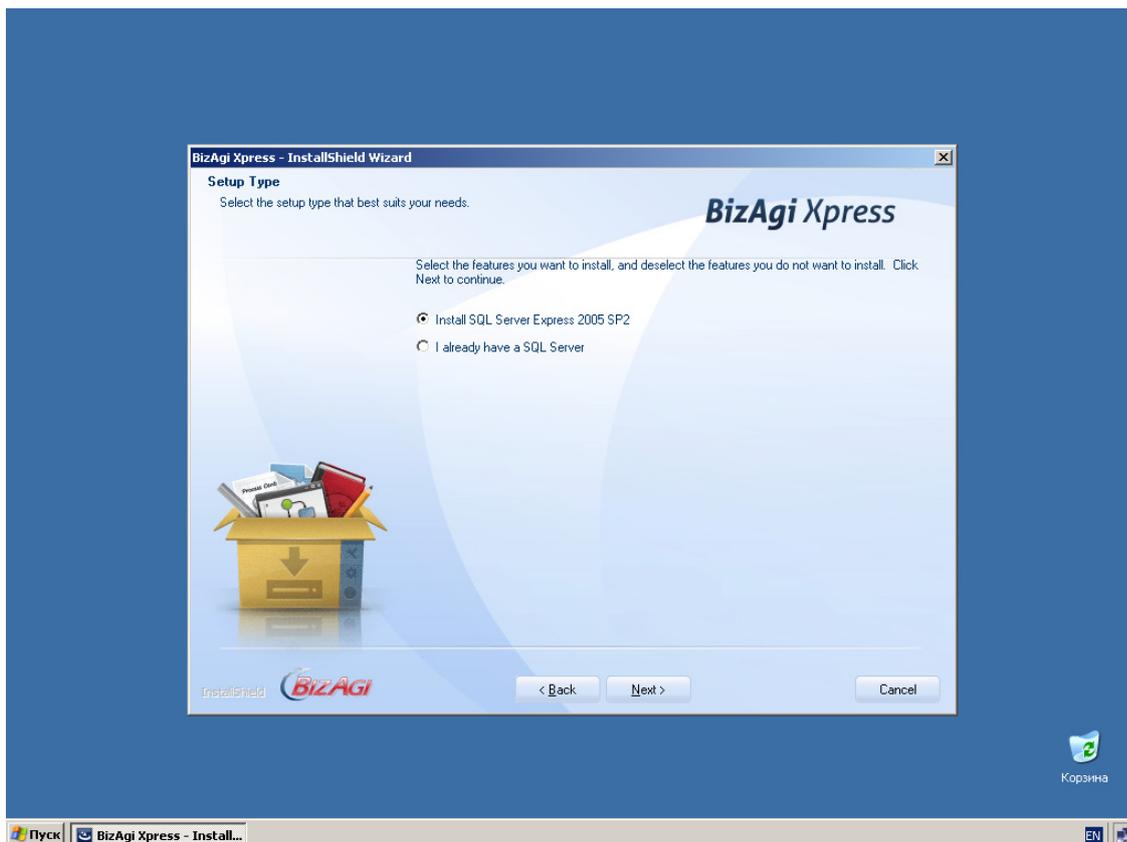


В этом случае продолжите установку после перезагрузки:

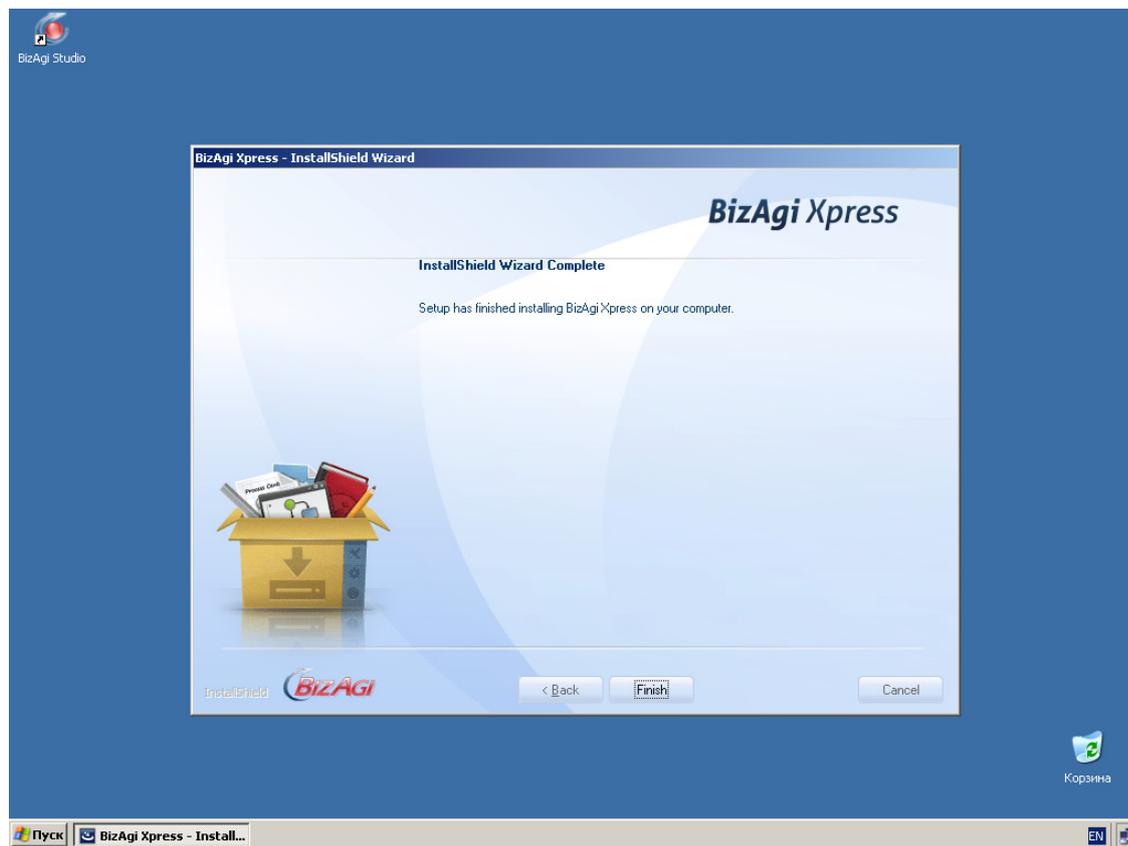




По умолчанию устанавливается бесплатная СУБД MS SQL Express:

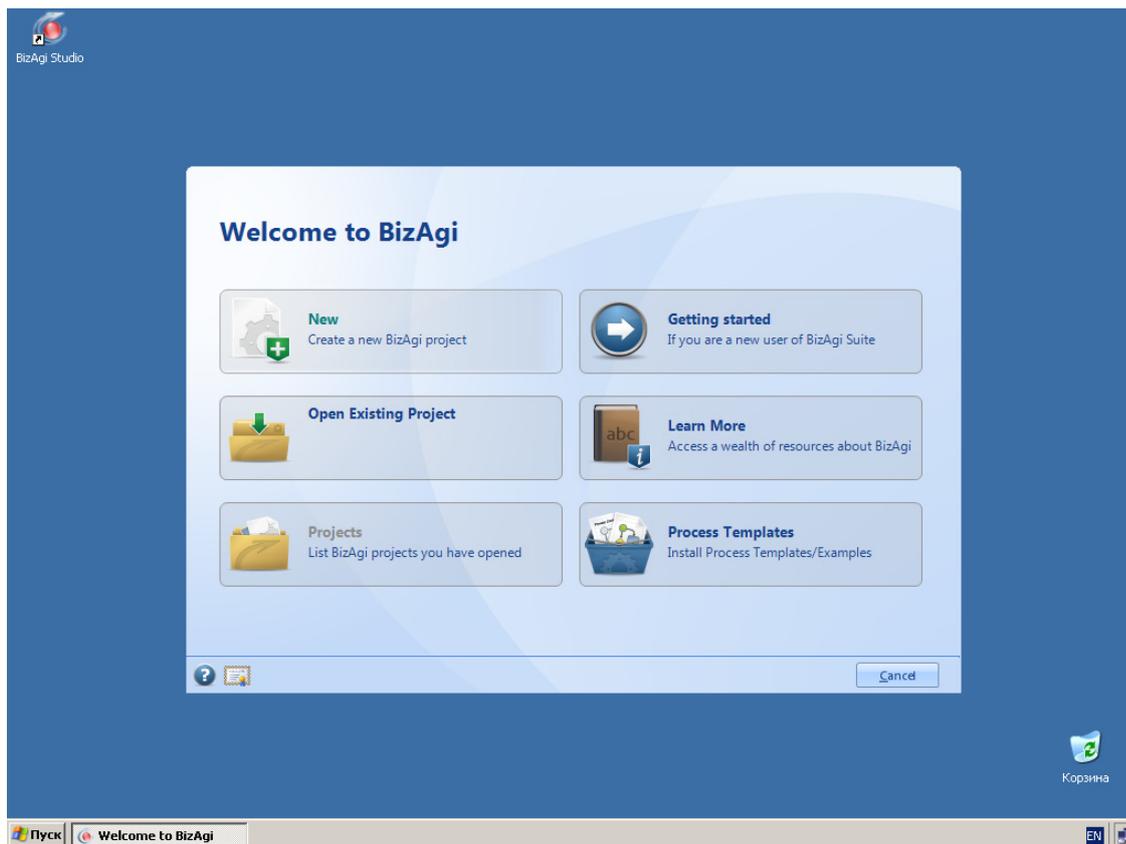


Общее время инсталляции составляет 10-15 минут.

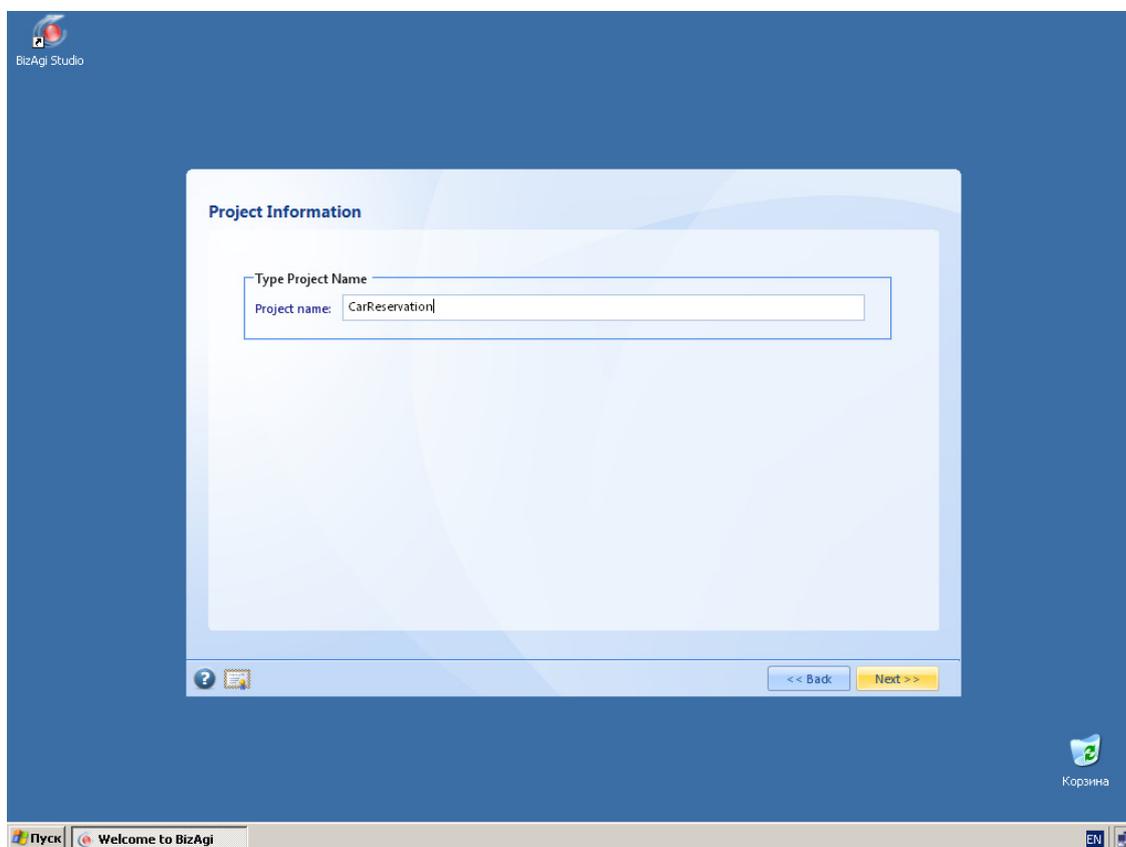


2. Создание проекта

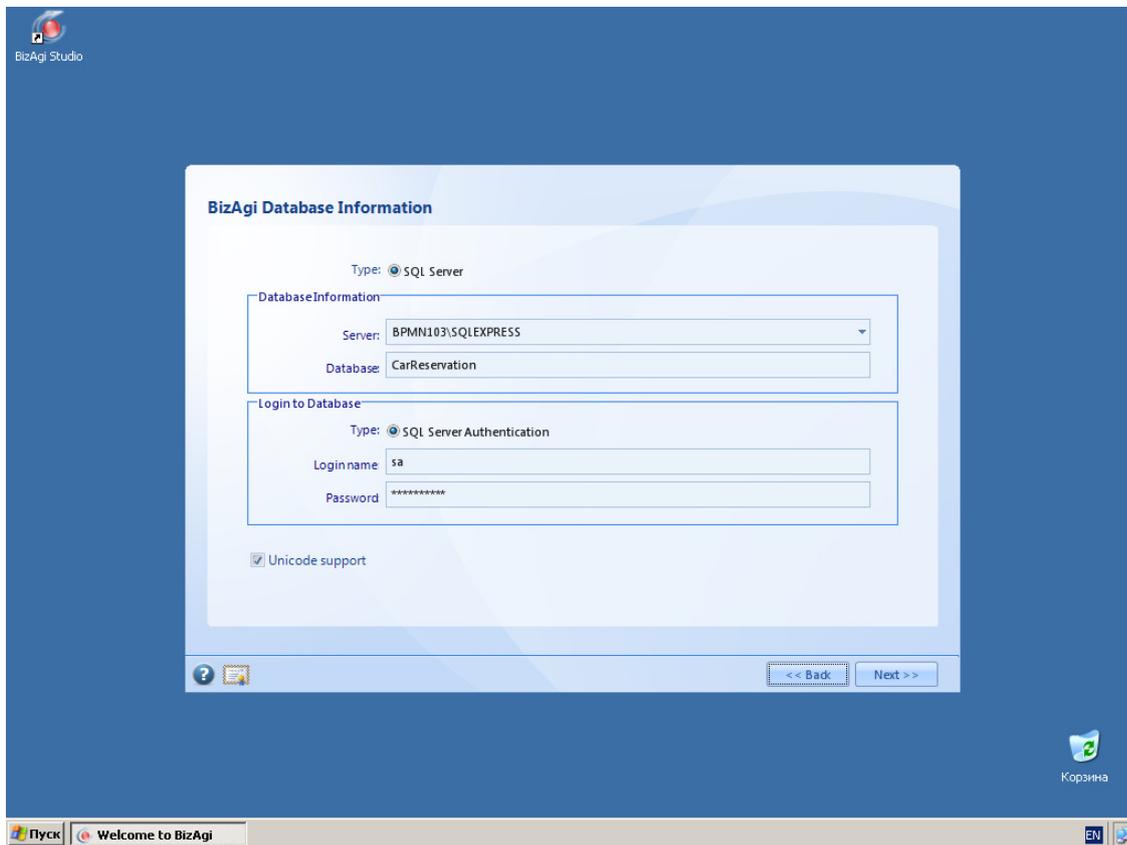
После завершения установки запустите BizAgi Studio:



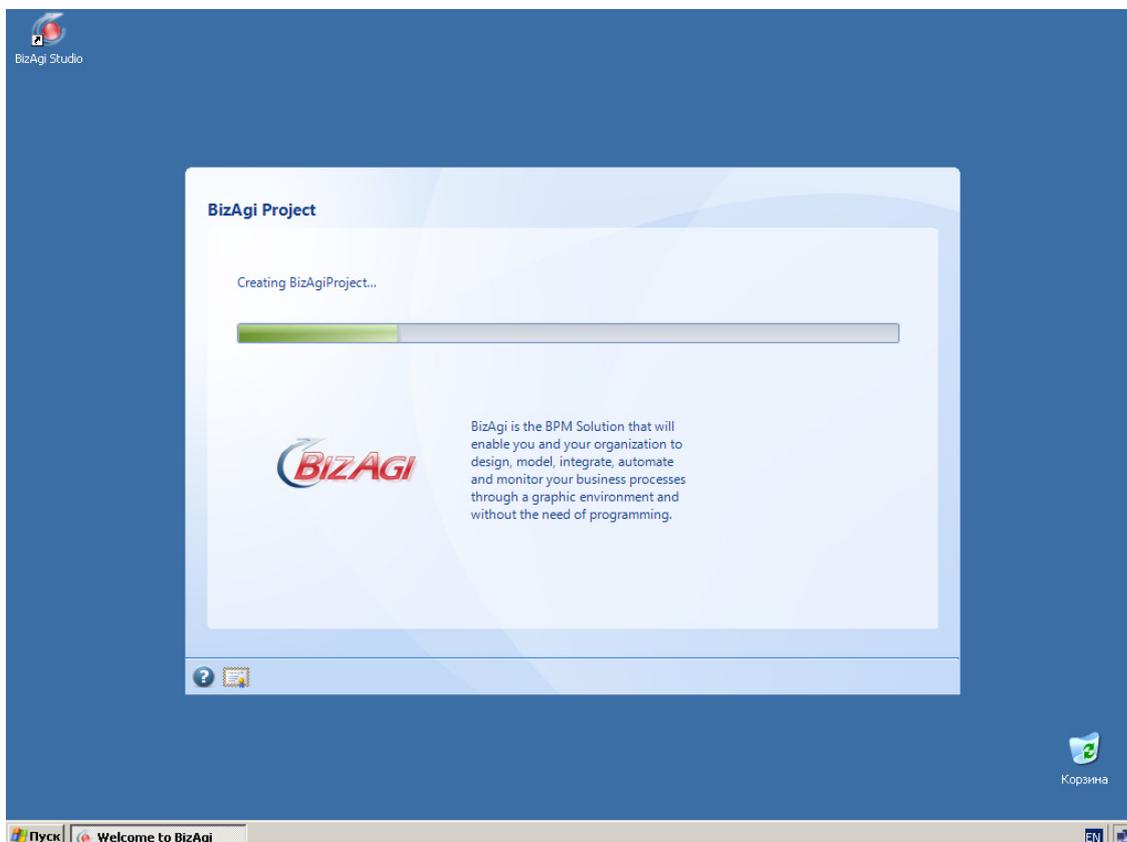
Создайте новый проект CarReservation:



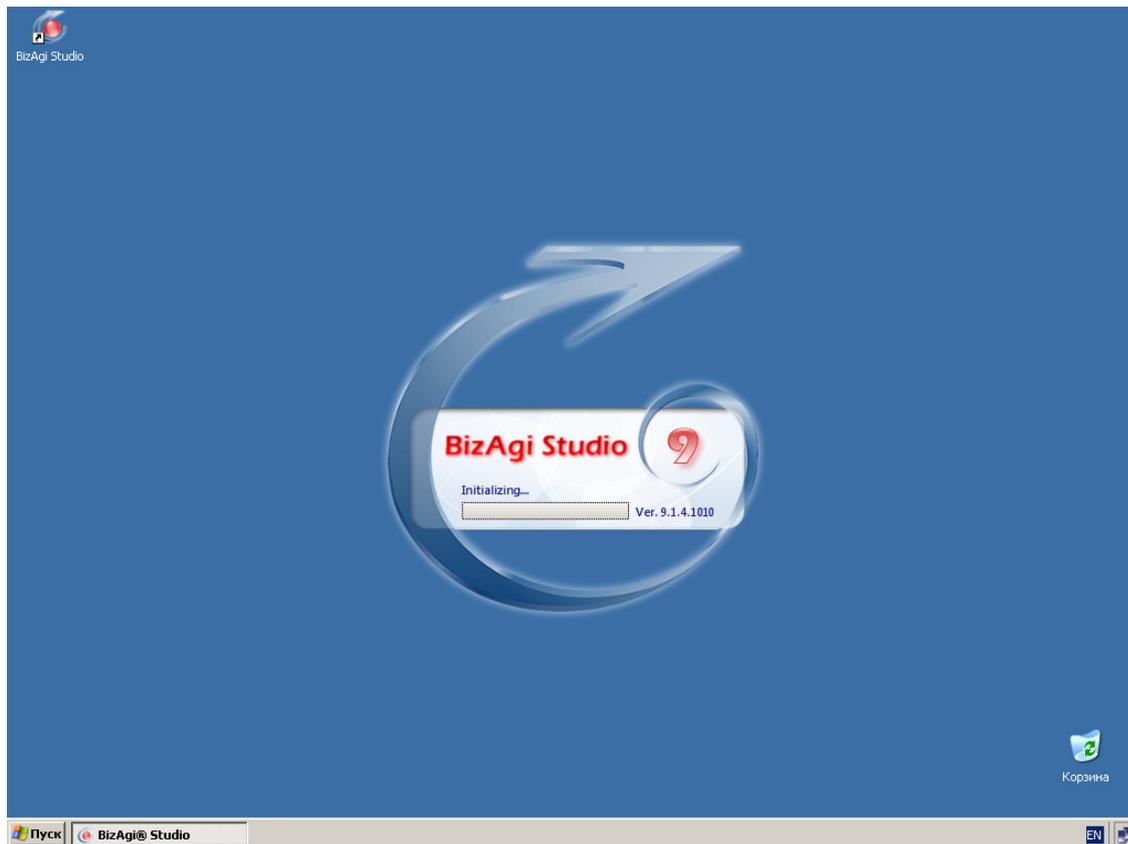
Программа запросит параметры подключения к базе данных. Если вы выбрали установку MS SQL Express, то ничего менять не надо. Если вы решили использовать уже имеющуюся у вас базу данных, то укажите параметры подключения к ней.



Под каждый проект BizAgI создает отдельную базу данных, имя базы данных совпадает с названием проекта. Проект в случае BizAgI – это вся система в целом, а не ее фрагмент, как у других BPM-вендоров. Вы можете создать отдельный проект для экспериментов или демонстрации, но в последствии вся ваша промышленная система будет одним проектом.



Когда проект создан, появляется заставка...

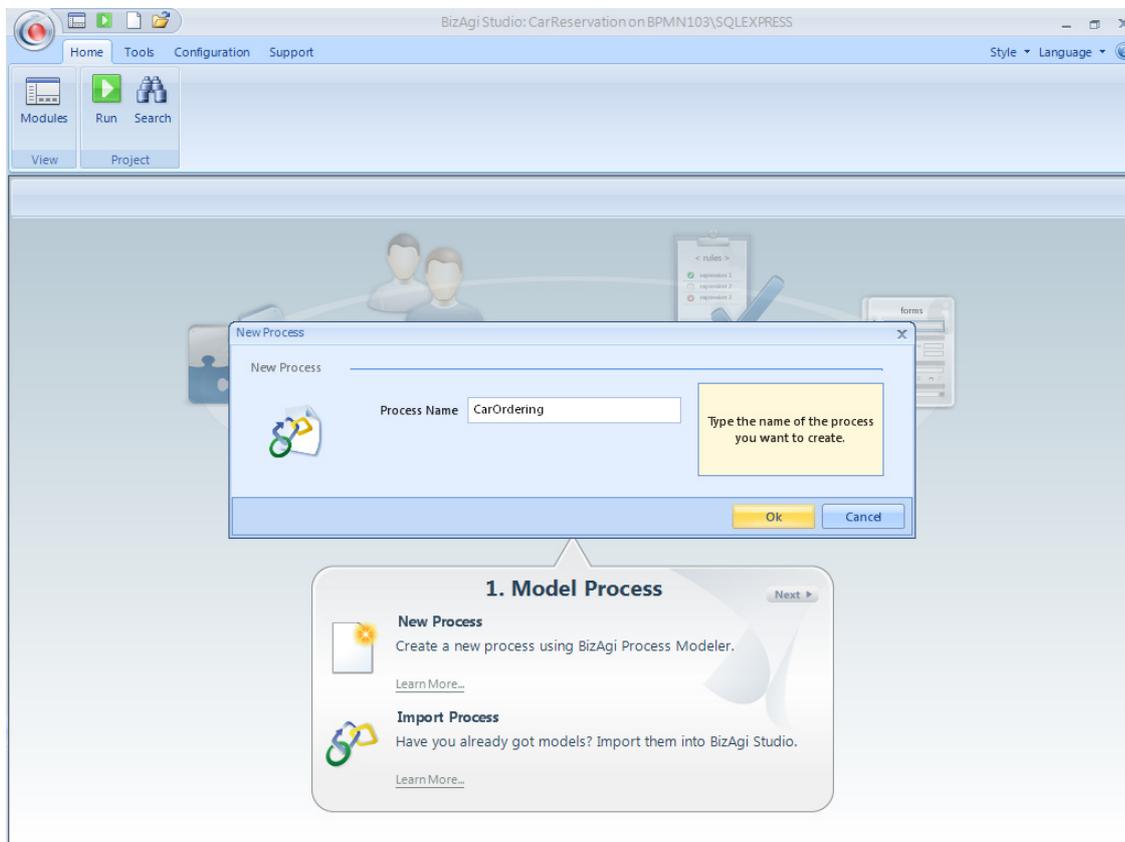


а вслед за заставкой – фирменная «карусель», которая наглядно демонстрирует последовательность шагов создания исполняемого бизнес-процесса:

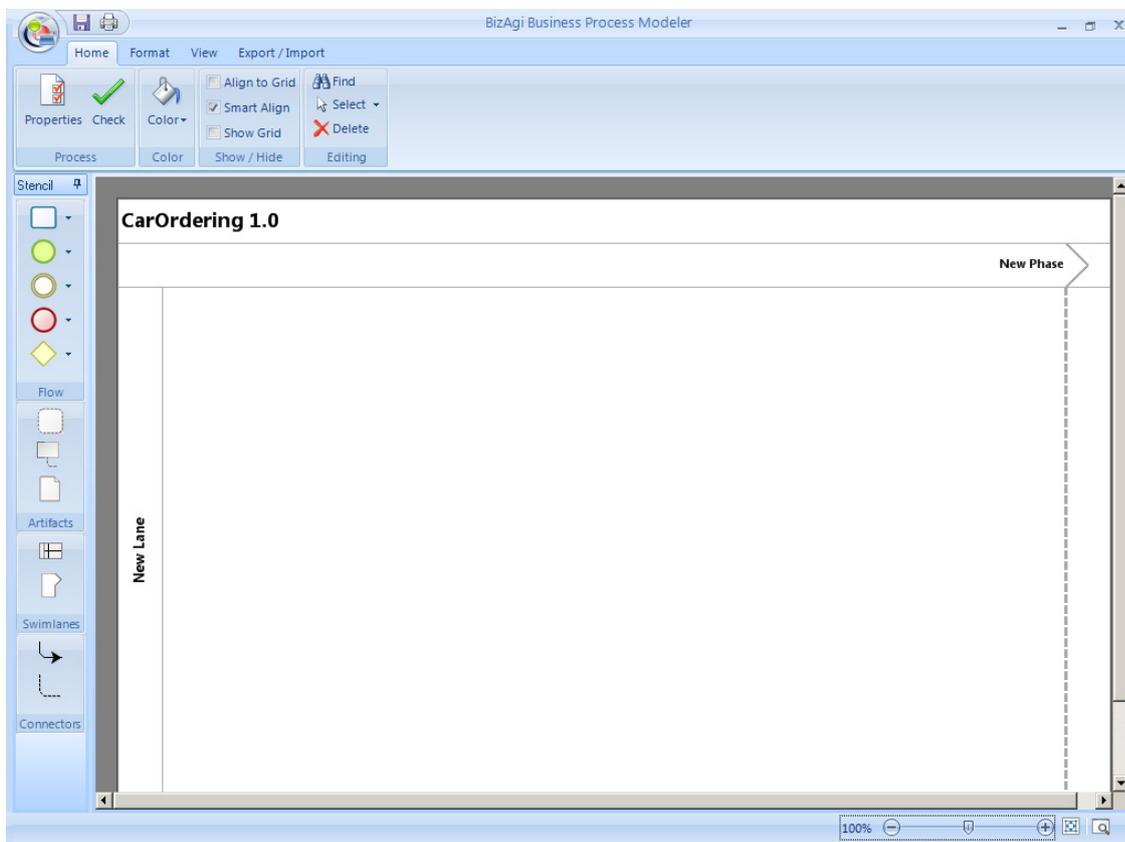


3. Схема процесса

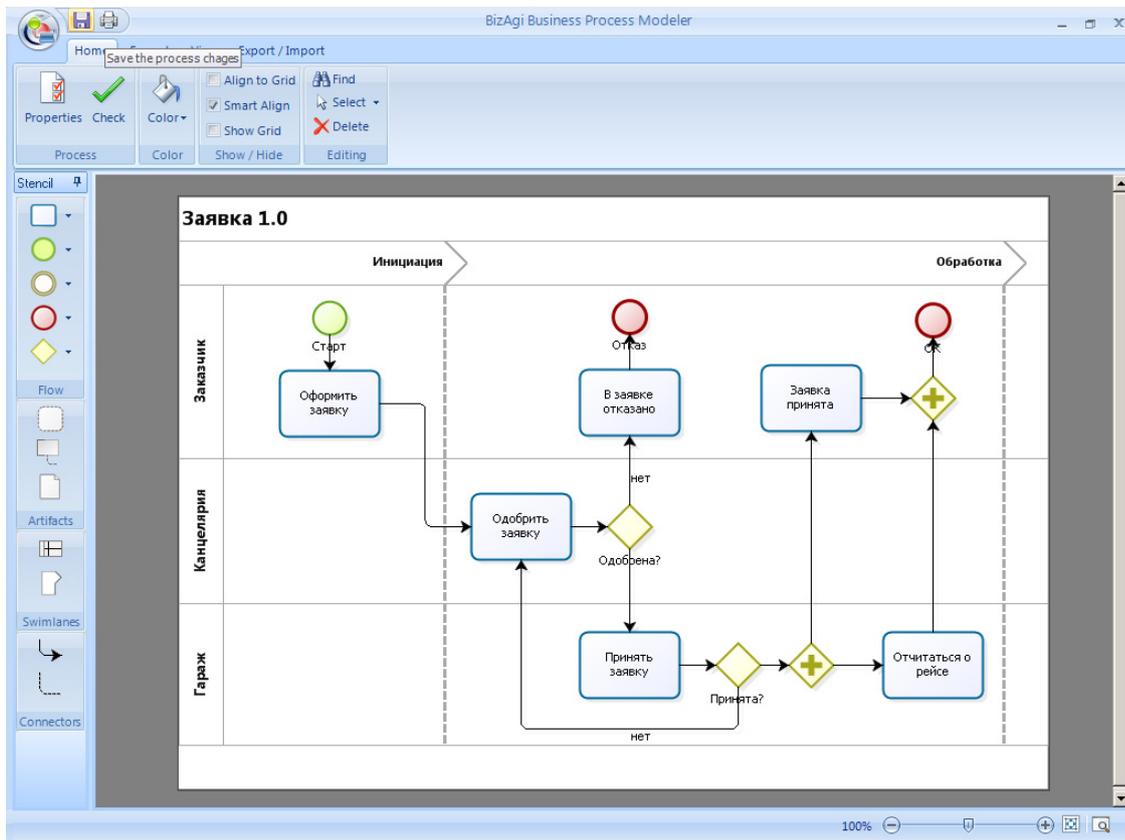
Первый шаг – моделирование процесса. Большинство объектов в BizAgі имеют название (Name) и название для отображение на экране (Display Name). Первые должны быть на латинице, вторые можно задавать на любых языках (в случае многоязычного приложения – на нескольких). Создайте новый процесс с названием «CarOrdering». Впоследствии вы сможете дать ему экранное название по-русски.



На экране появится пустая диаграмма:



Вы можете рисовать схему процесса с нуля, экспортировать диаграмму из BizAgі Modeler (файлы с расширением .bpm) или диаграмму в формате XPDL. Создайте схему процесса:



Закончив работу, выйдите из моделиера, сохранив результат, и нажмите кнопку «Next», чтобы перейти к следующему шагу:

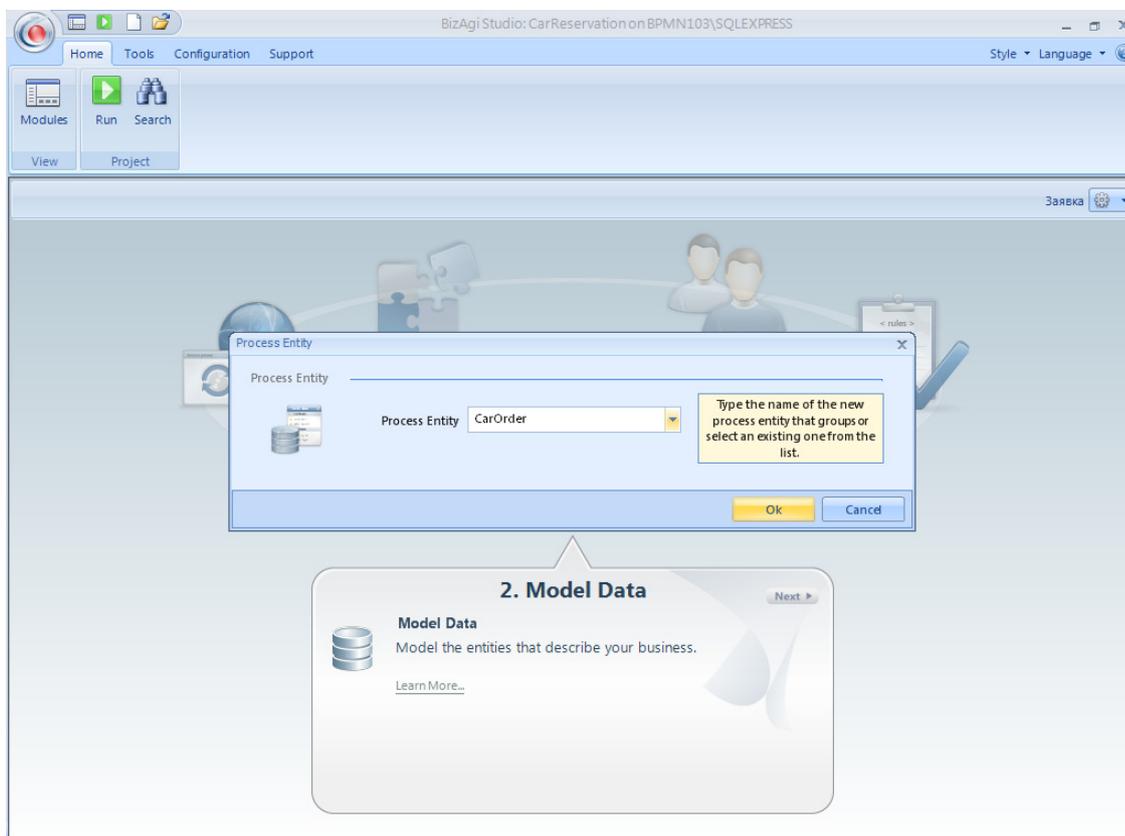


4. Схема данных

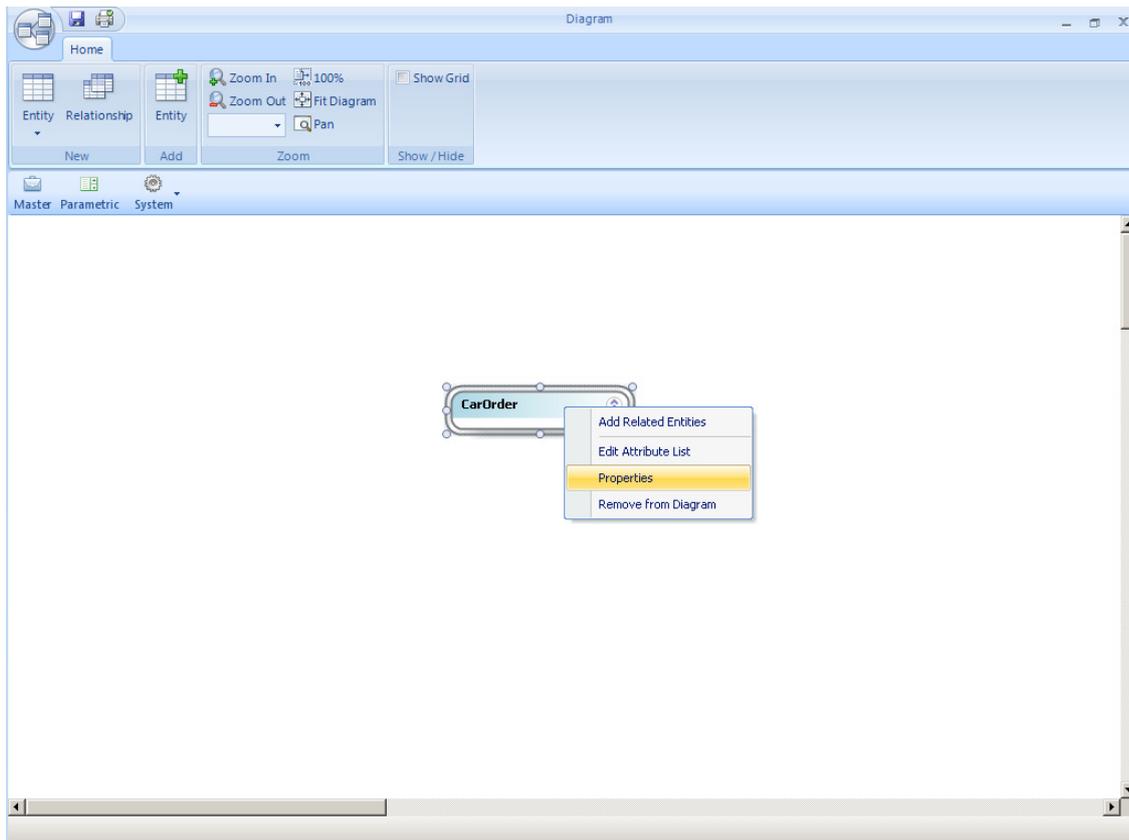
Следующий шаг – моделирование данных процесса:



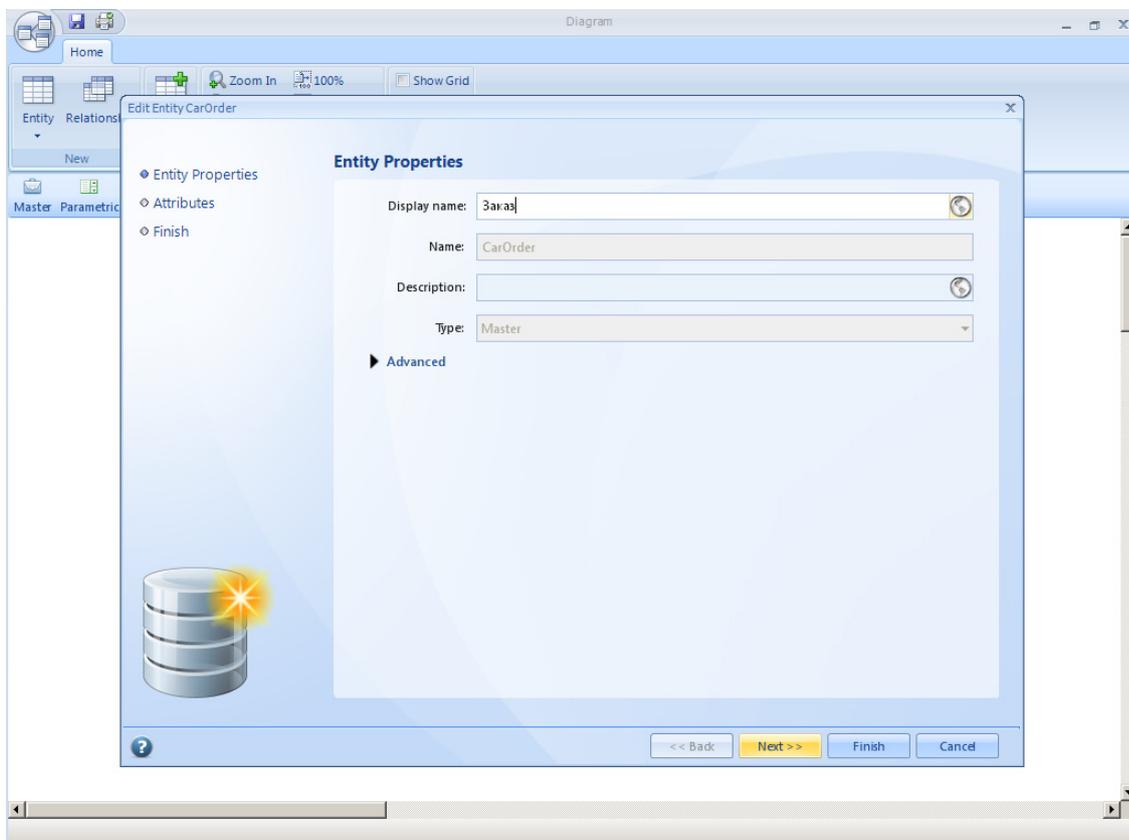
Первое что вам необходимо сделать – это задать имя таблицы, в которой будут храниться данные экземпляров процессов – каждому экземпляру процесса будет соответствовать одна запись в этой таблице. Как и в случае процесса, название должно задавать латиницей: «CarOrder».



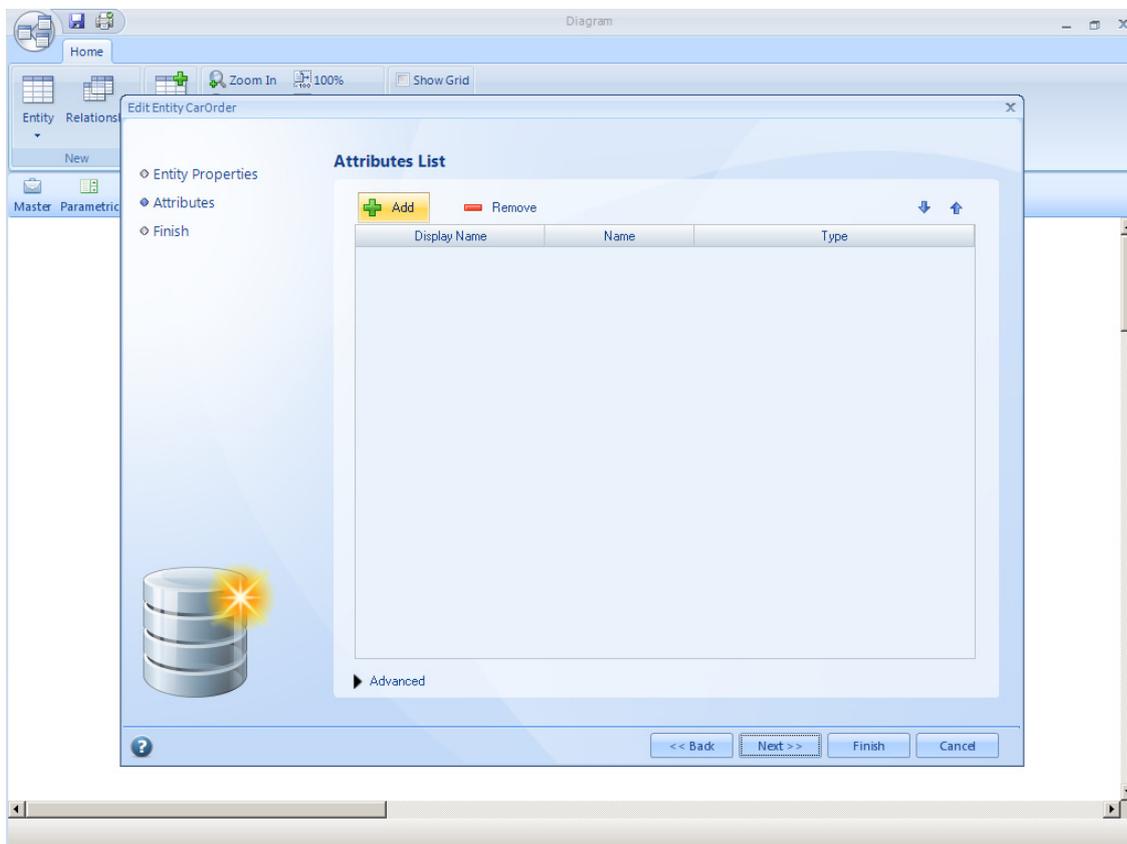
На экране появится диаграмма данных, вначале она состоит только из одного объекта. Щелкните правой кнопкой и выберите «Properties», чтобы задать название объекта по-русски и определить его атрибуты:



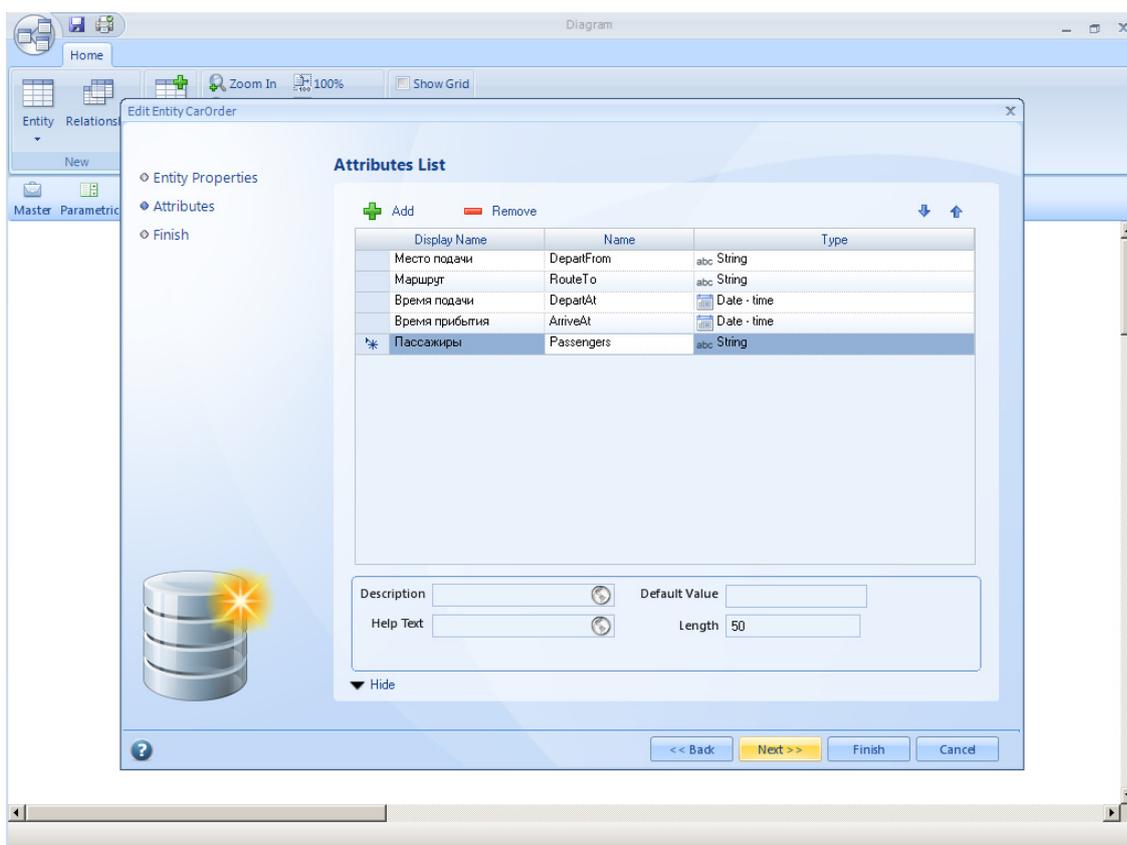
Задайте название объекта и щелкните «Next»:



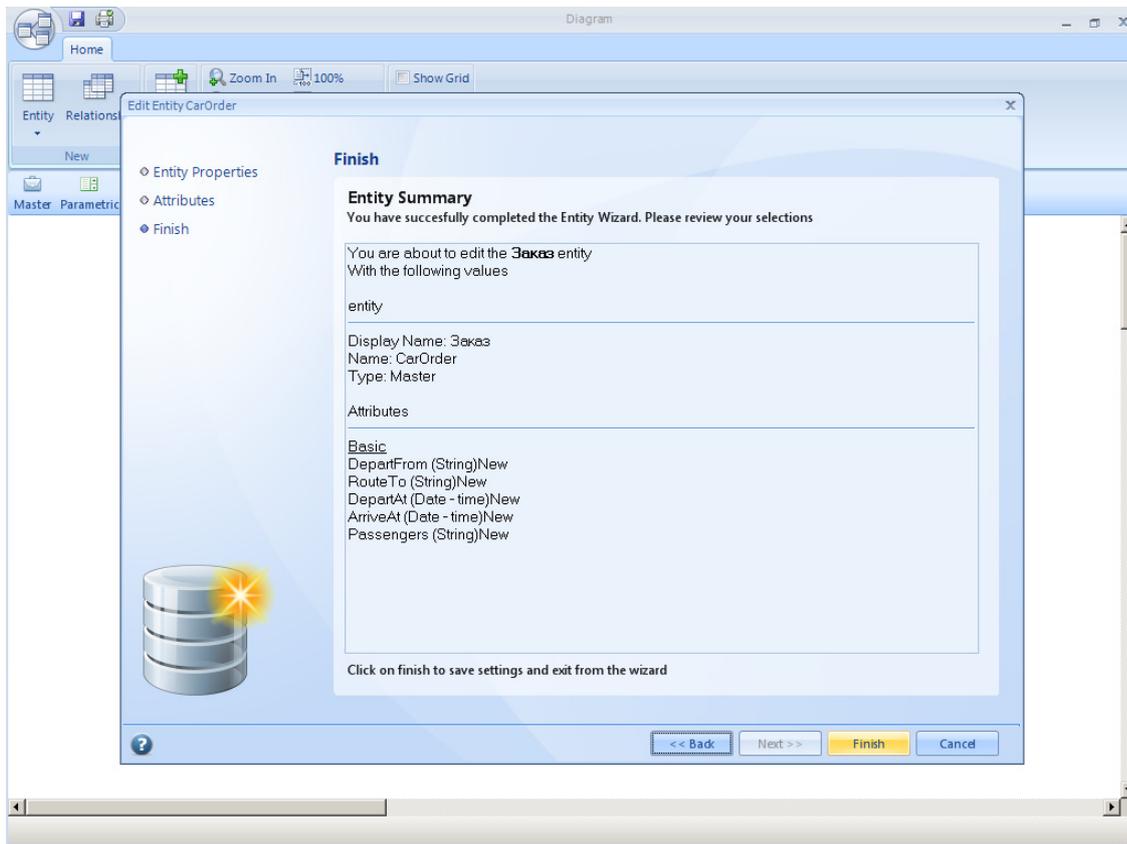
На следующем экране задаются атрибуты. Щелкните «Add», чтобы добавить атрибут:



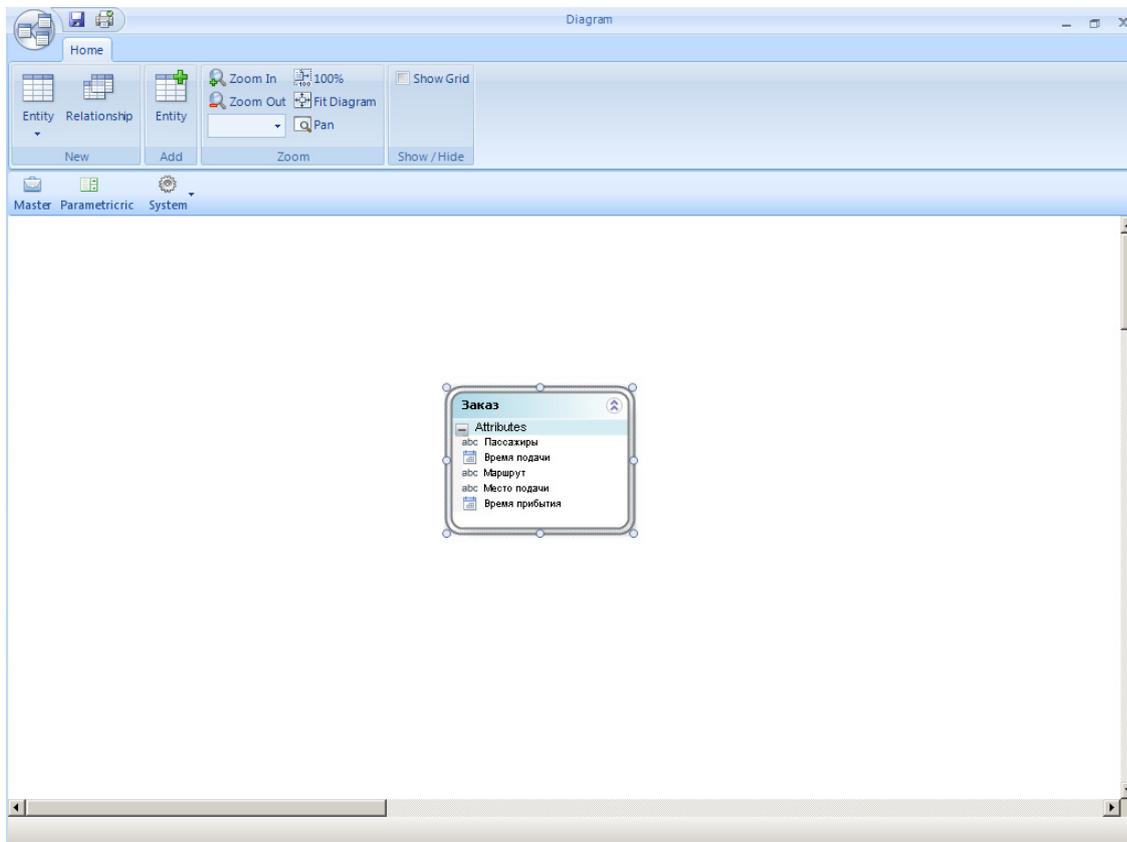
Введите атрибуты заявки:



Закончив работу, кликните «Finish»:



Вернувшись на диаграмму, вы увидите объект и его атрибуты (при желании их можно скрыть, щелкнув по двойной вертикальной стрелке). Сохраните диаграмму и закройте окно, чтобы вернуться в карусель.

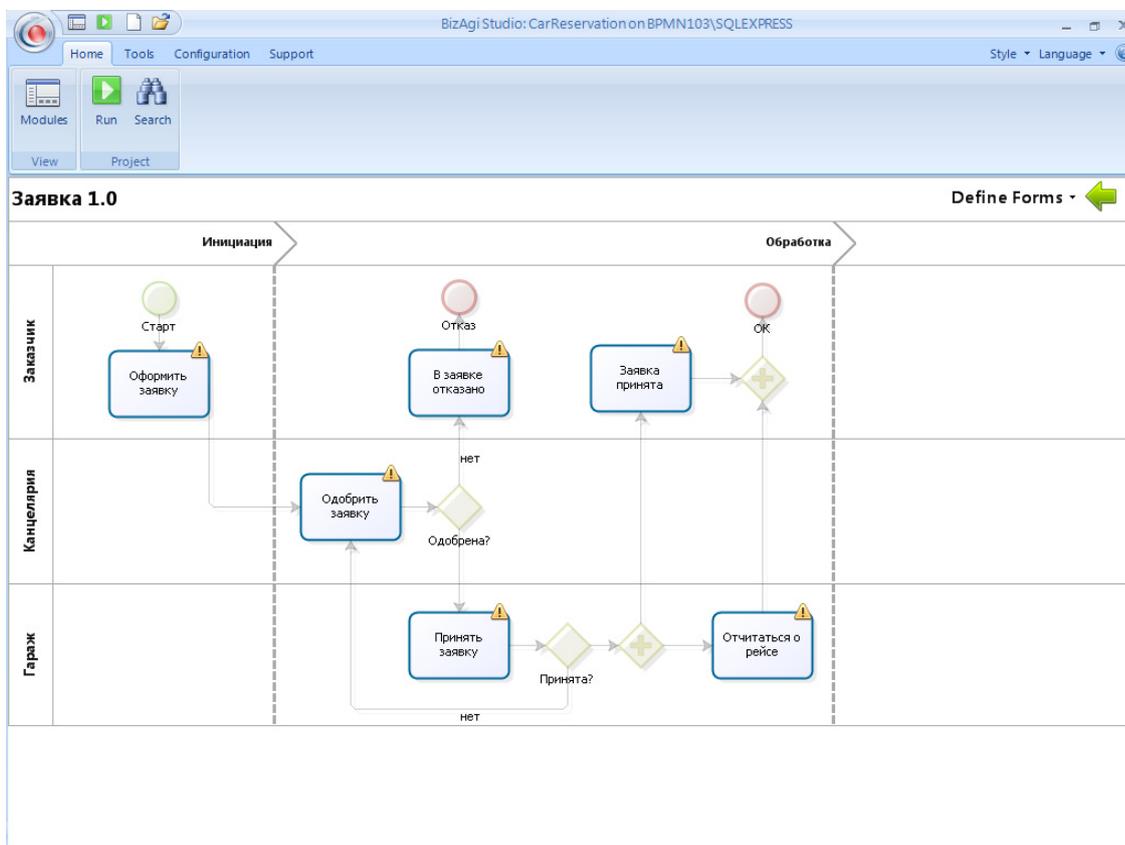


5. Формы

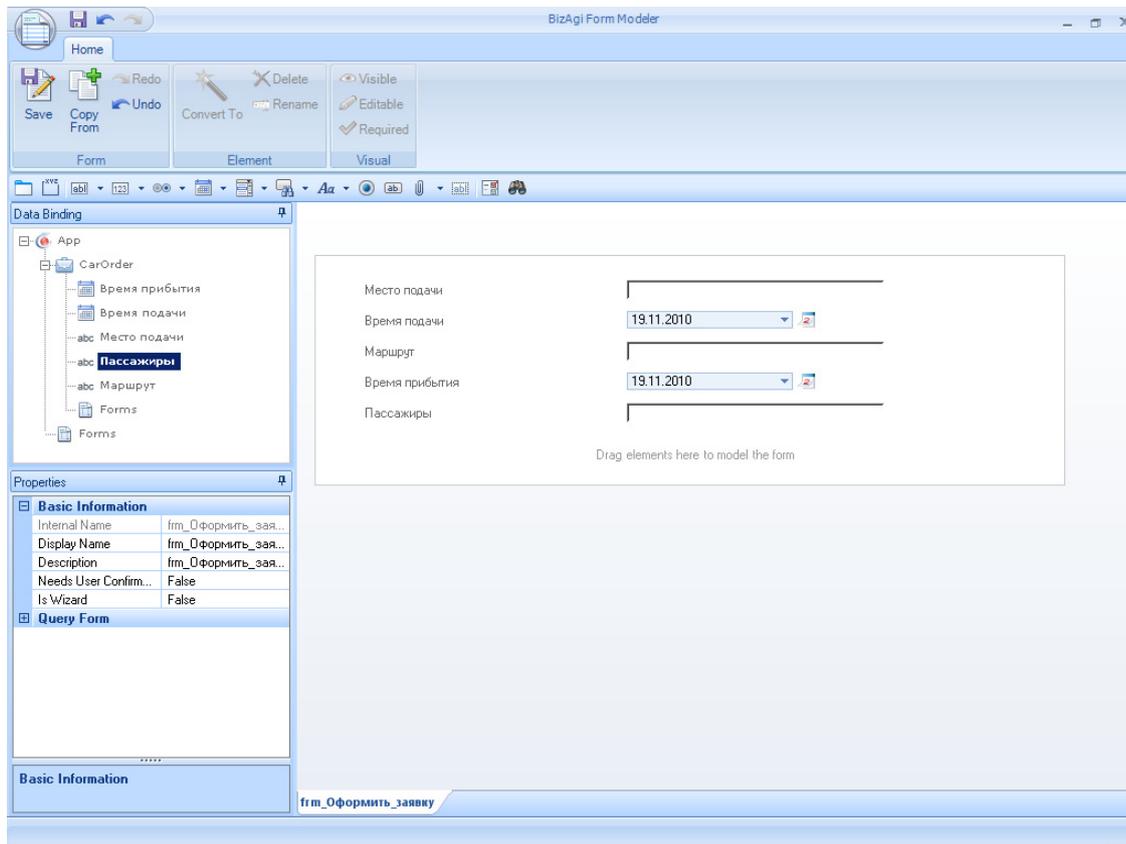
Теперь нарисуем формы пользовательского интерфейса к задачам процесса:



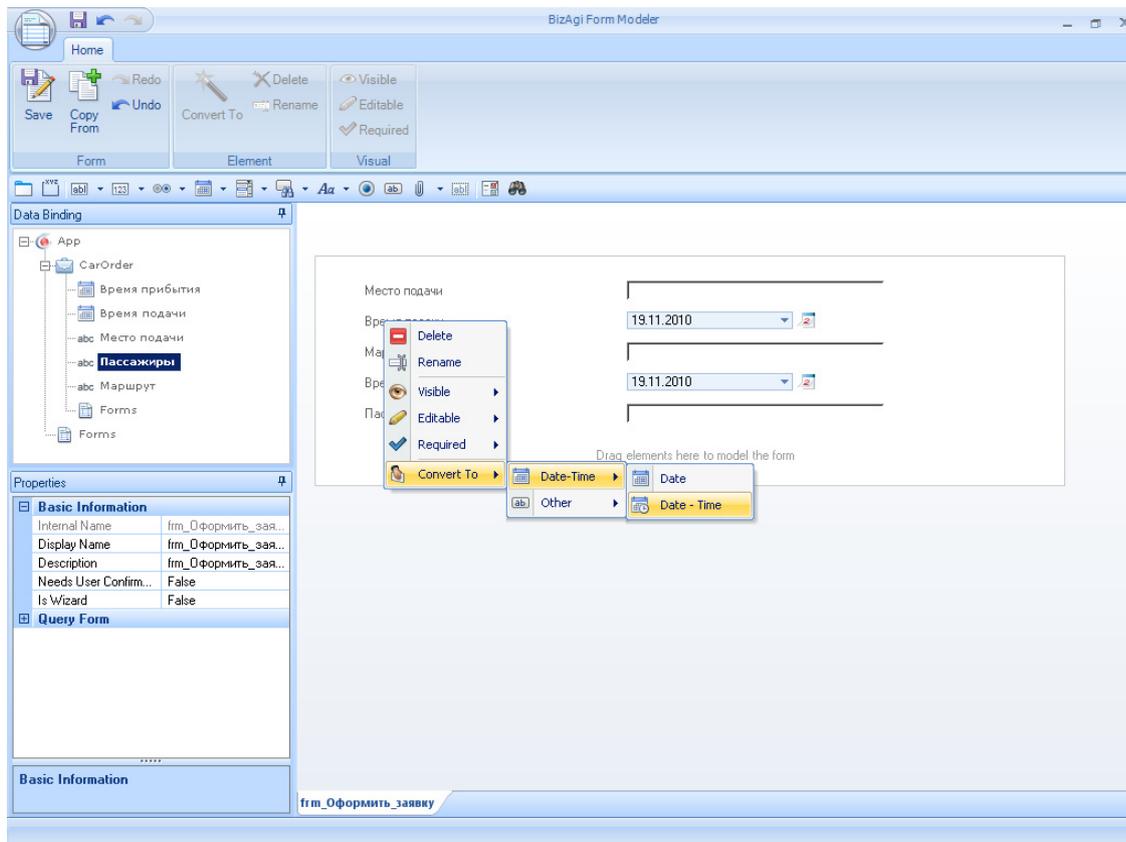
Желтым восклицательным знаком помечены задачи, для которых формы не определены. Щелкните по задаче «Оформить заявку»:



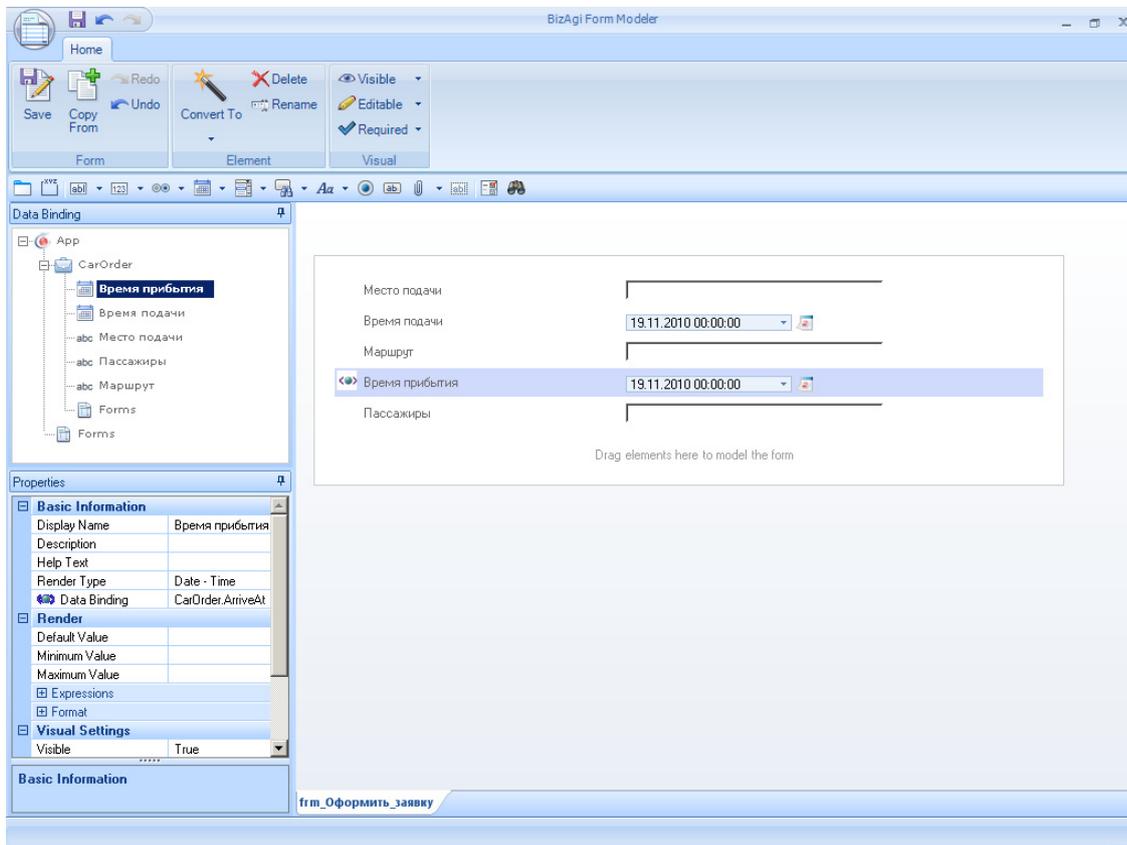
Перетащите из раскрывающегося списка слева на холст справа атрибуты заявки:



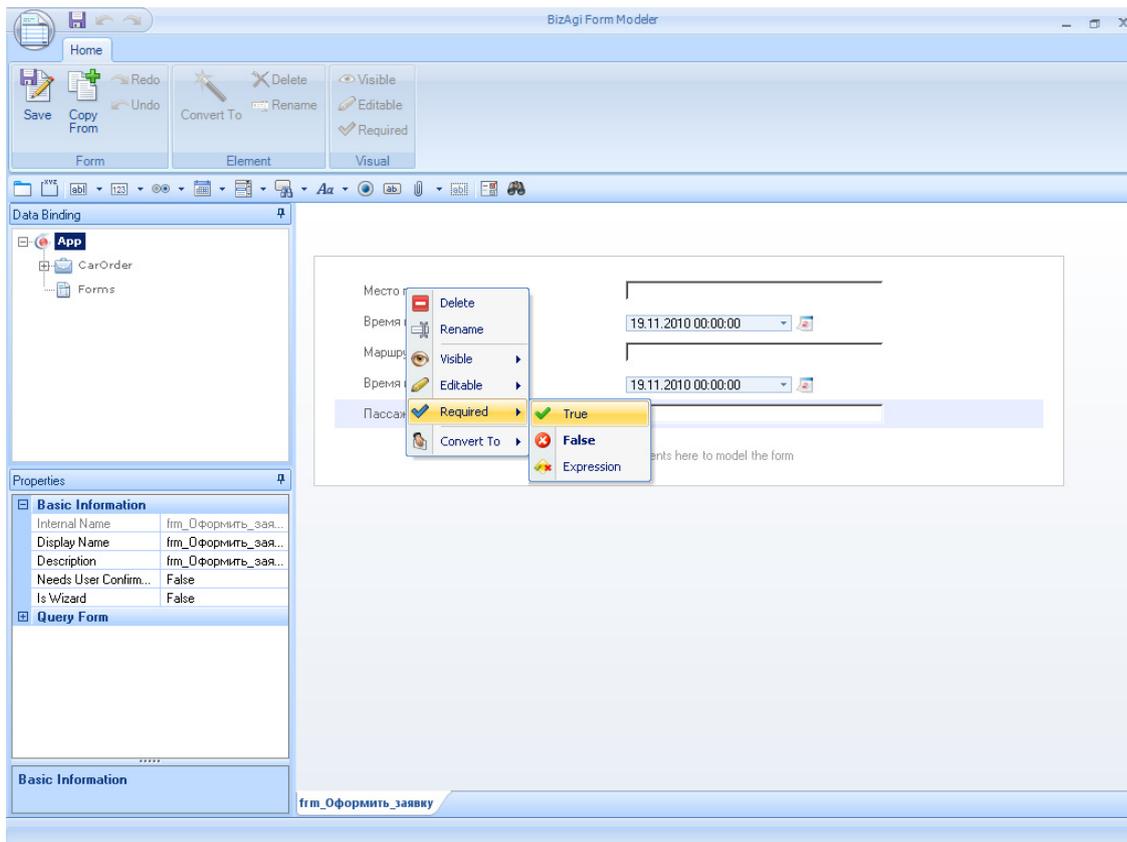
Для полей «Время подачи» и «Время прибытия» укажите тип данных «Date-Time»:



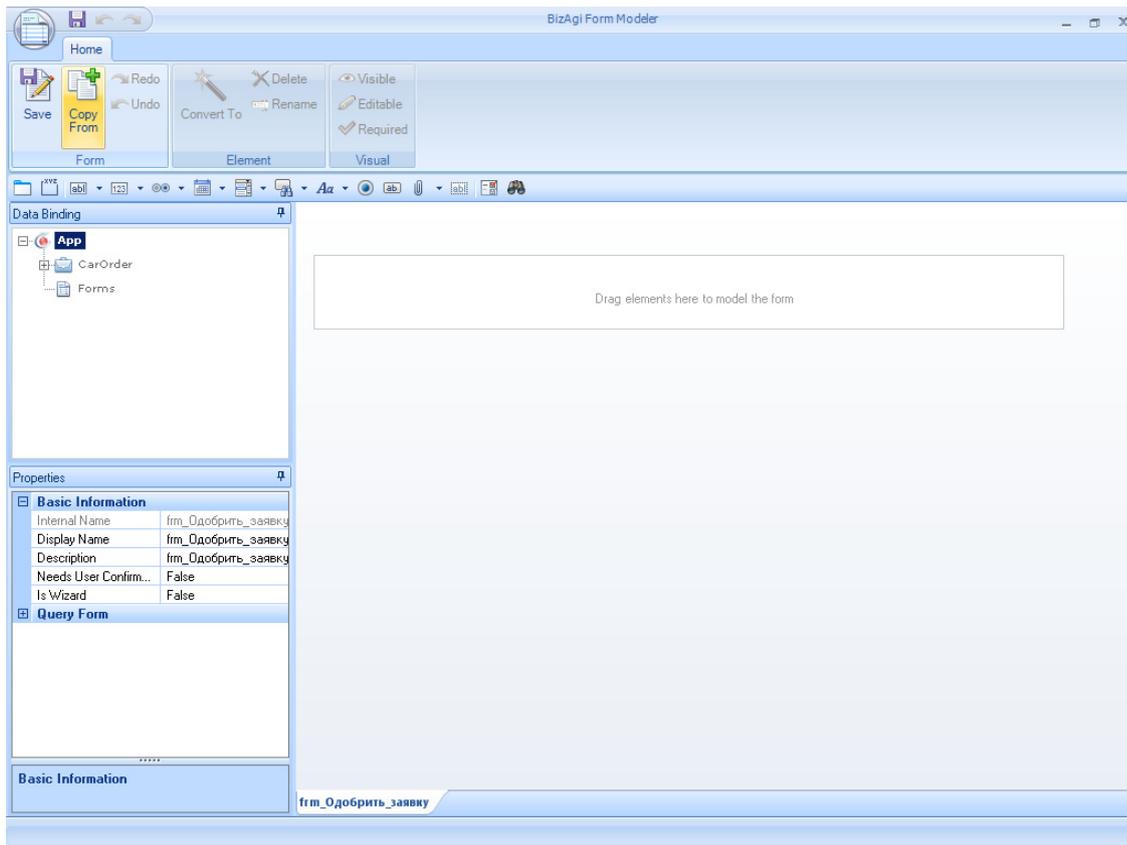
Вид поля для ввода в результате изменится:



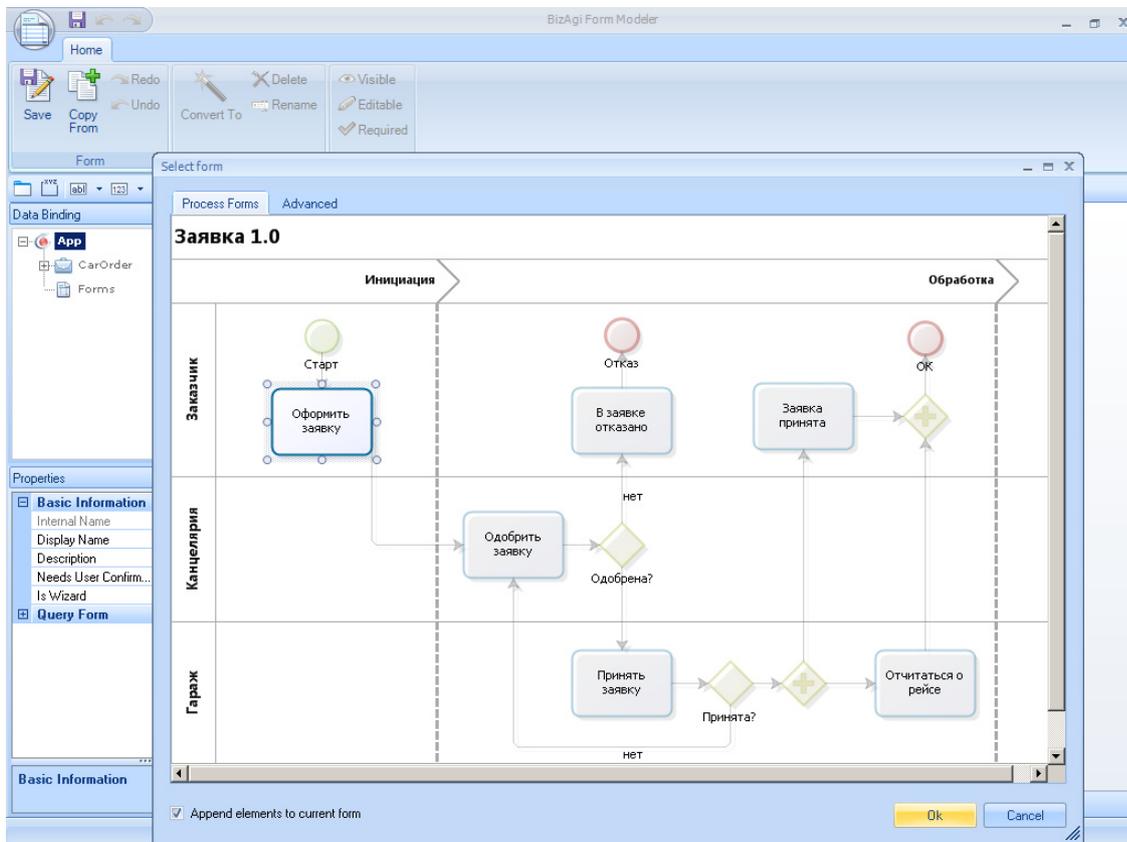
Сделайте поля «Место подачи» и «Время подачи» обязательными для ввода:



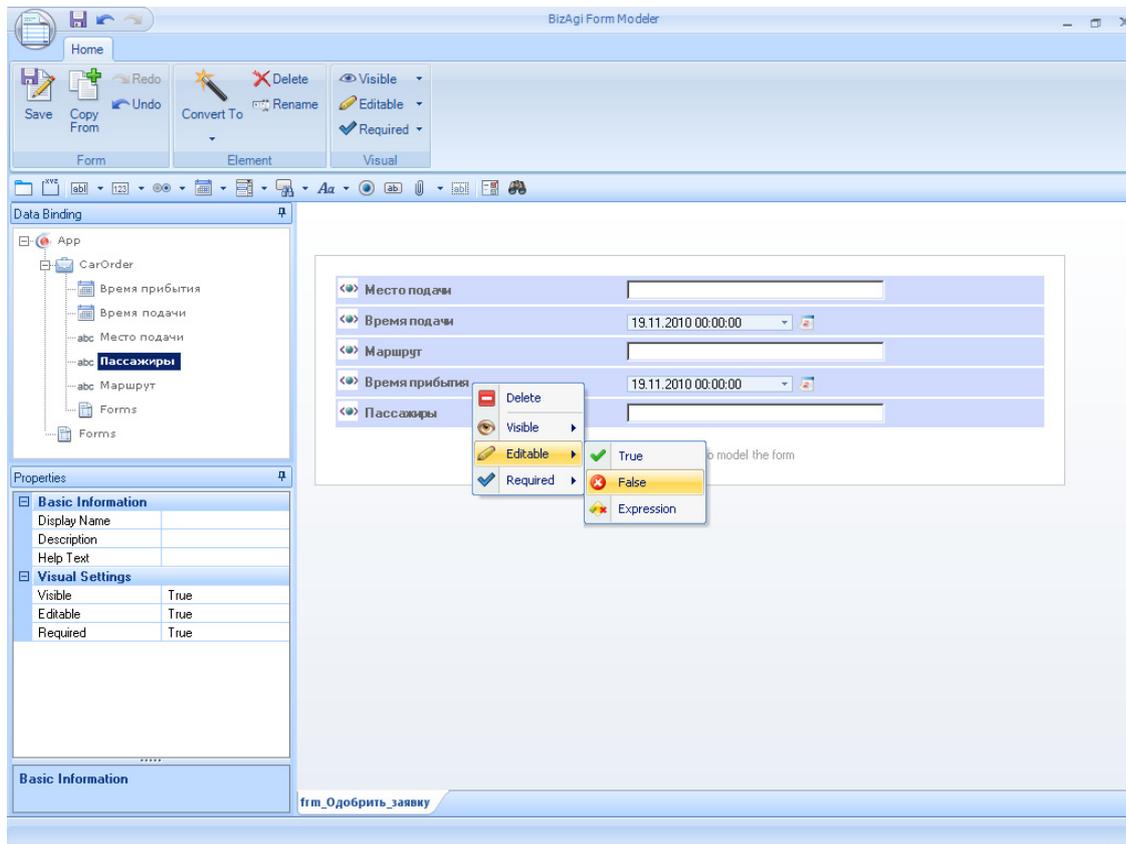
Сохраните форму и откройте форму для следующего шага «Одобрить заявку». Чтобы не рисовать форму с нуля, щелкните по кнопке «Copy From»:



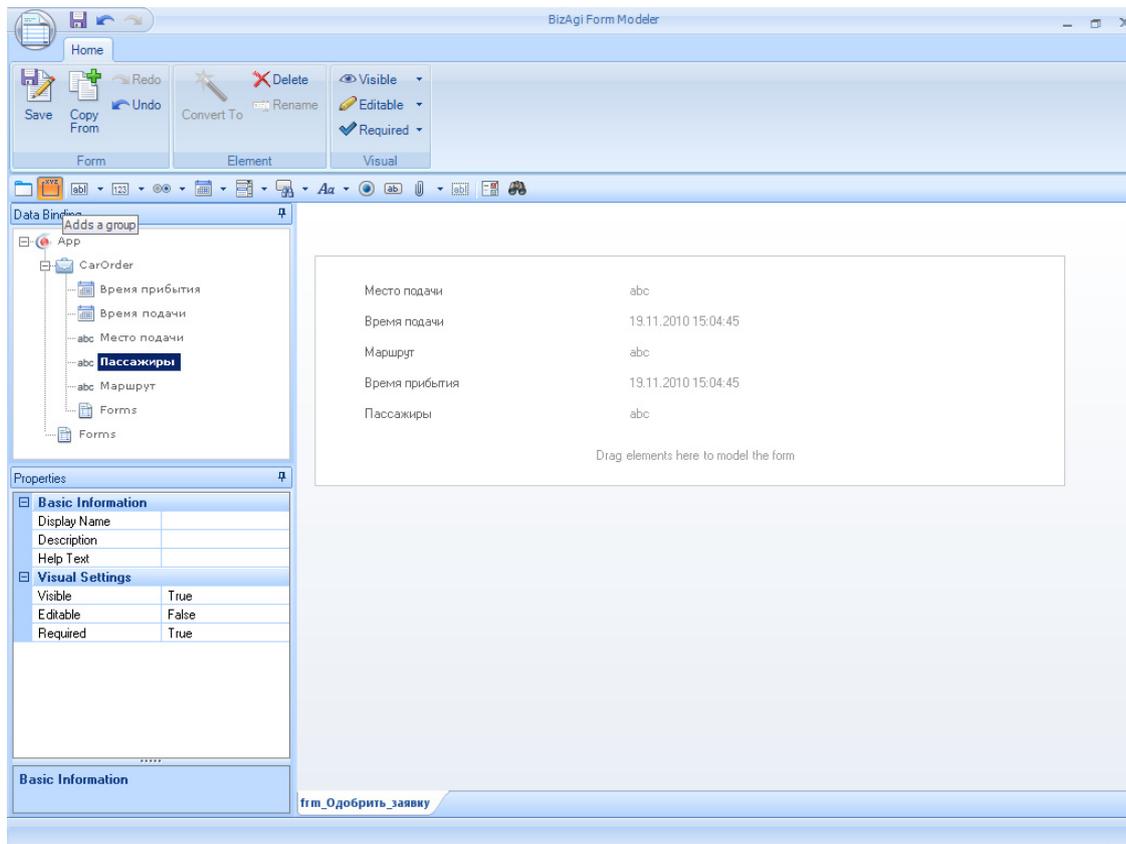
Скопируйте форму «Оформить заявку»:



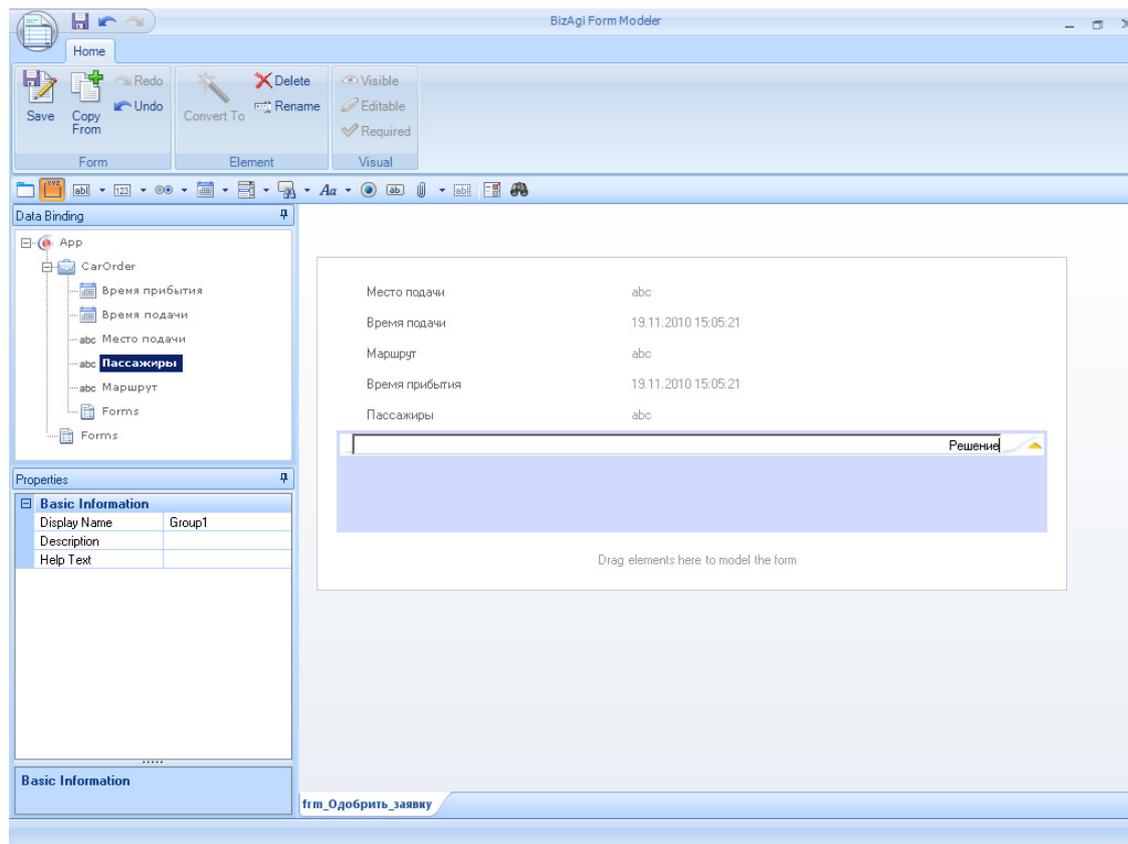
Поля скопировались, осталось только запретить их изменение:



Результат:

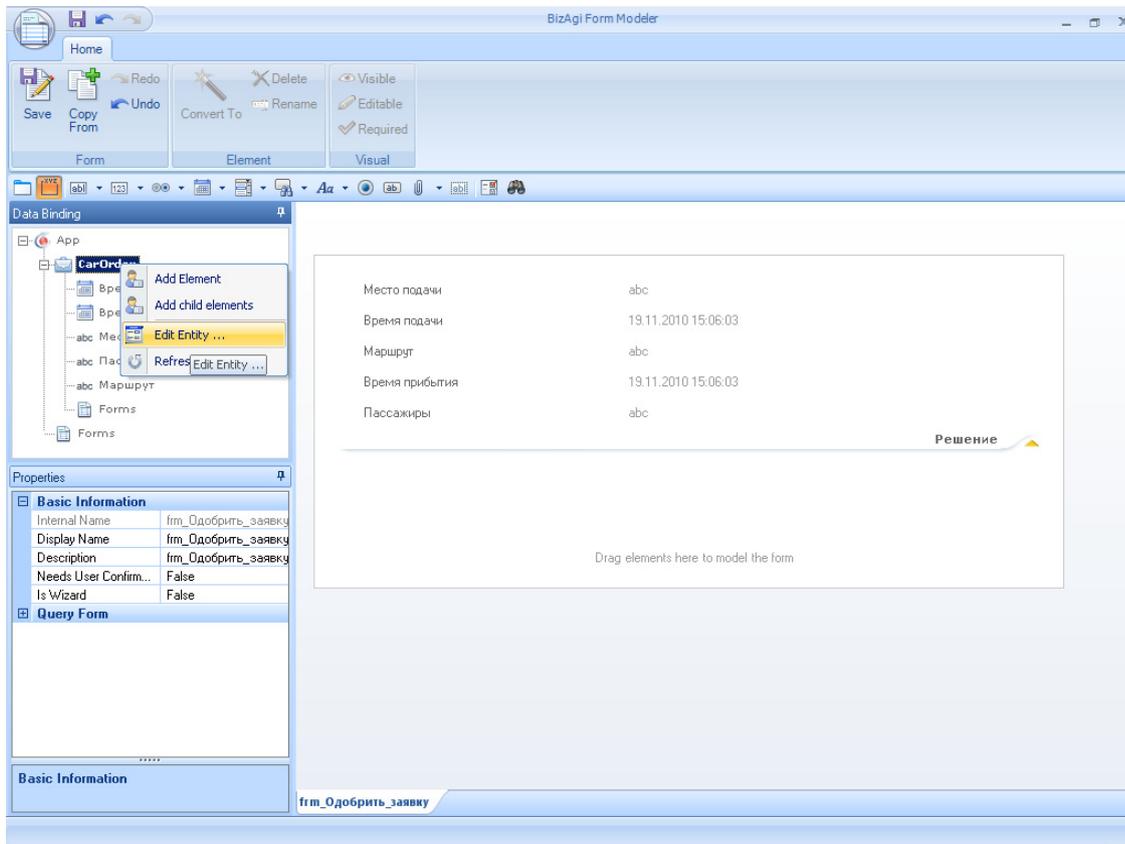


На этой форме секретарь (дорожка «Канцелярия» на диаграмме) одобряет или не одобряет заявку. Чтобы визуально разделить форму, перетащите из панели инструментов на форму элемент «Group» и назовите группу «Решение»:

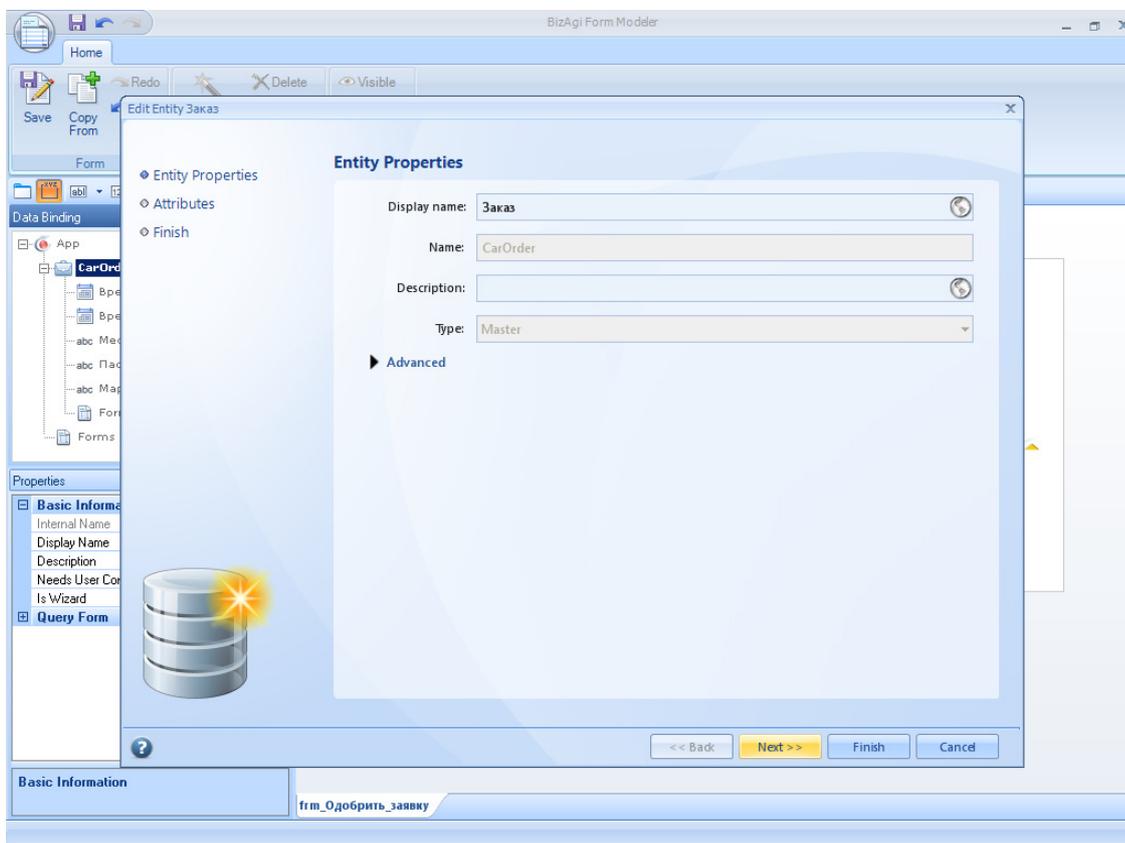


6. Добавление атрибутов на лету

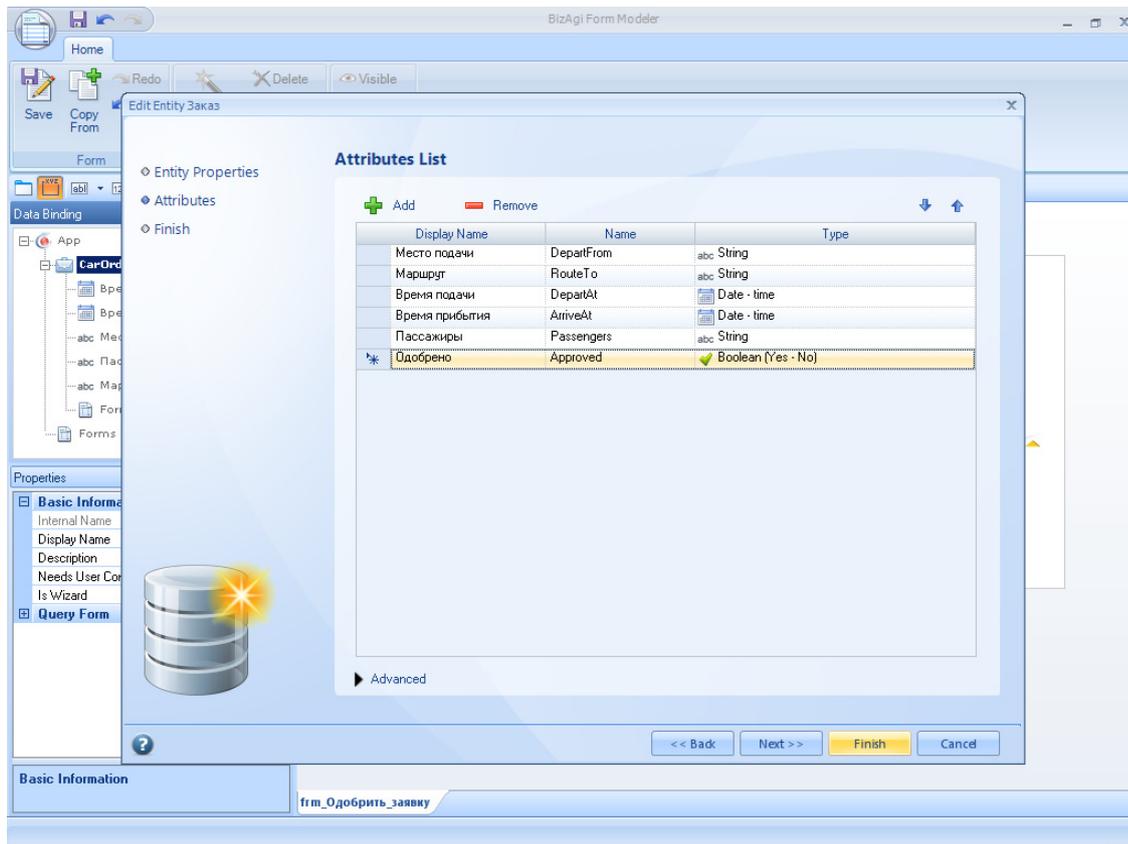
Но кажется, мы забыли определить атрибут для хранения решения секретаря. Не беда, это можно сделать прямо из редактора форм. Щелкните правой кнопкой по объекту «CarOrder» и выберите пункт «Edit Entity...»



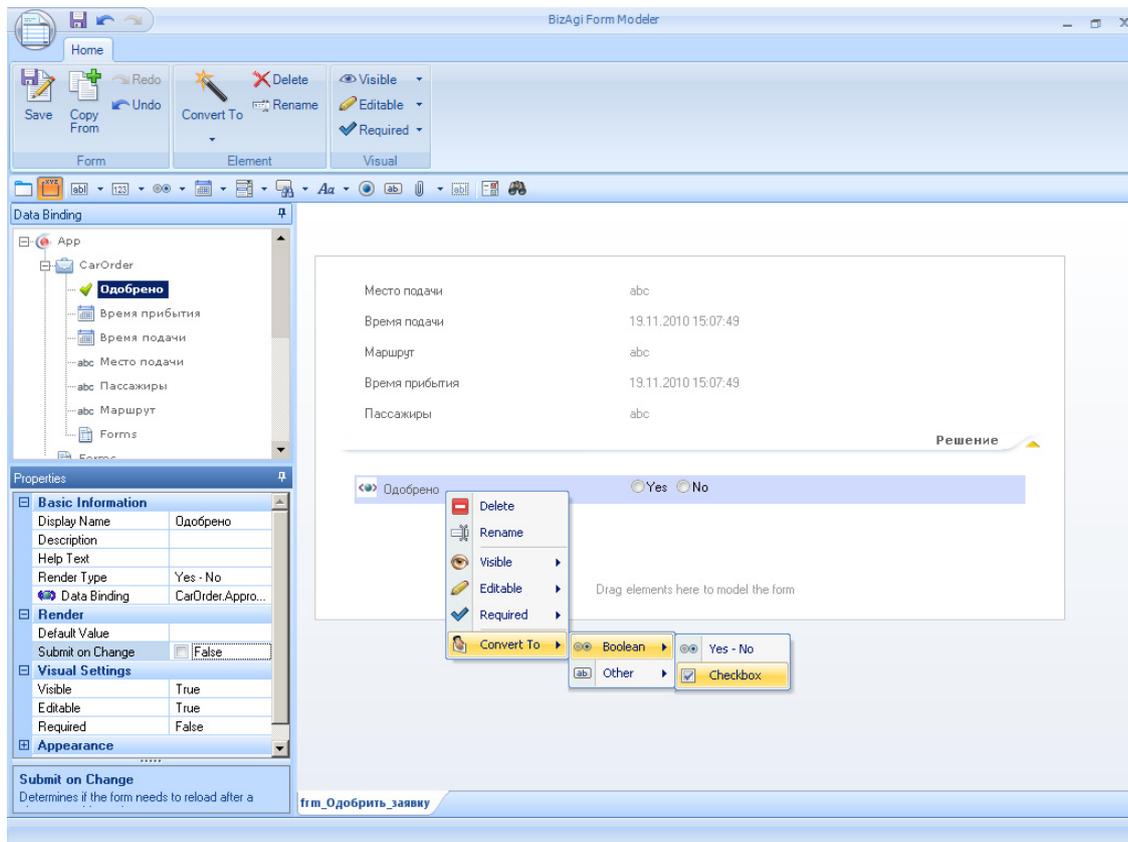
Вы попадете в ту же форму редактирования параметров и атрибутов объекта. Щелкните «Next», чтобы перейти к атрибутам:



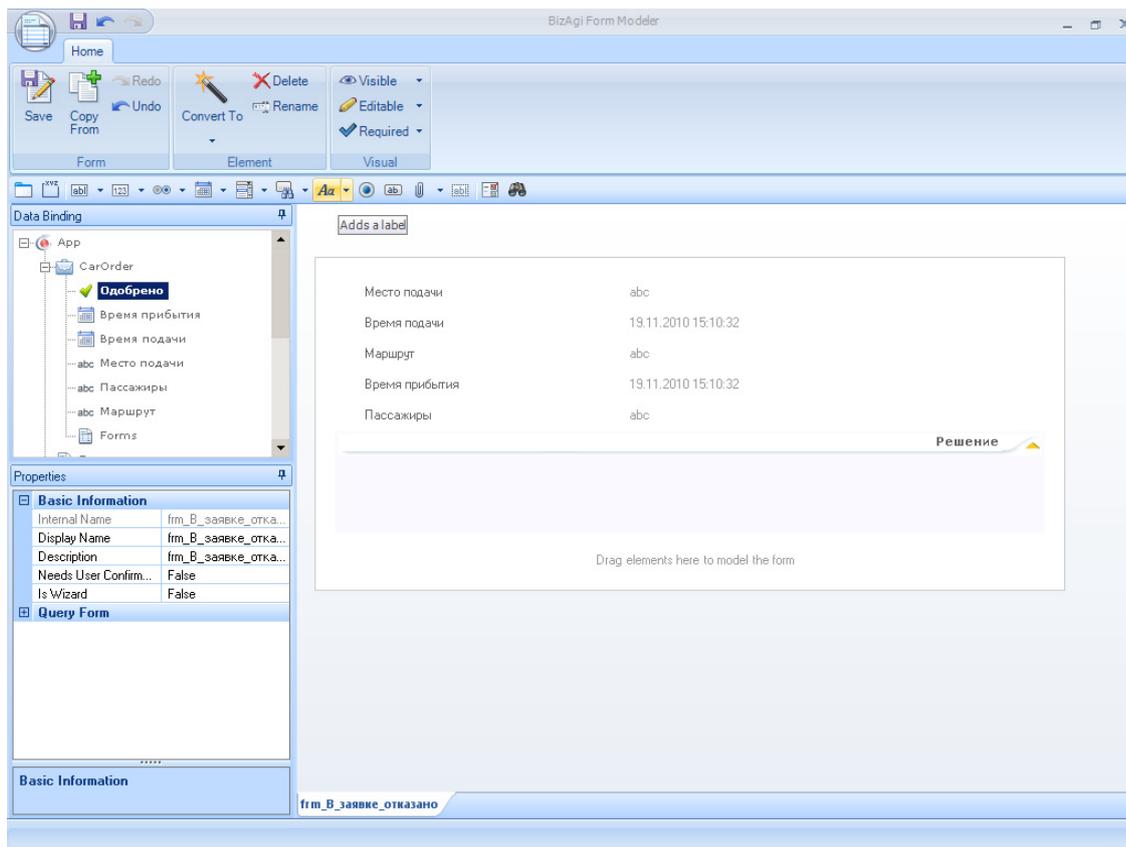
Добавьте булев атрибут «Одобрено»:



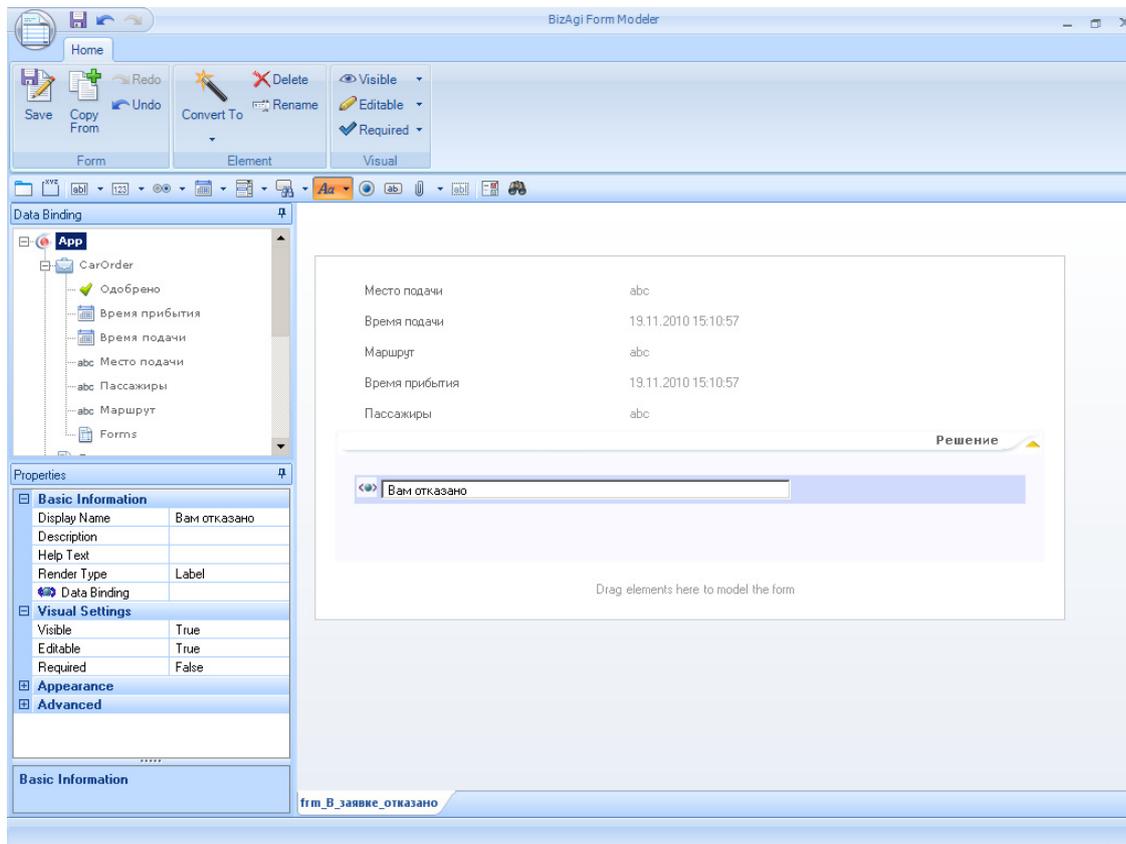
Его визуальное представление при желании можно сменить с radio button на checkbox:



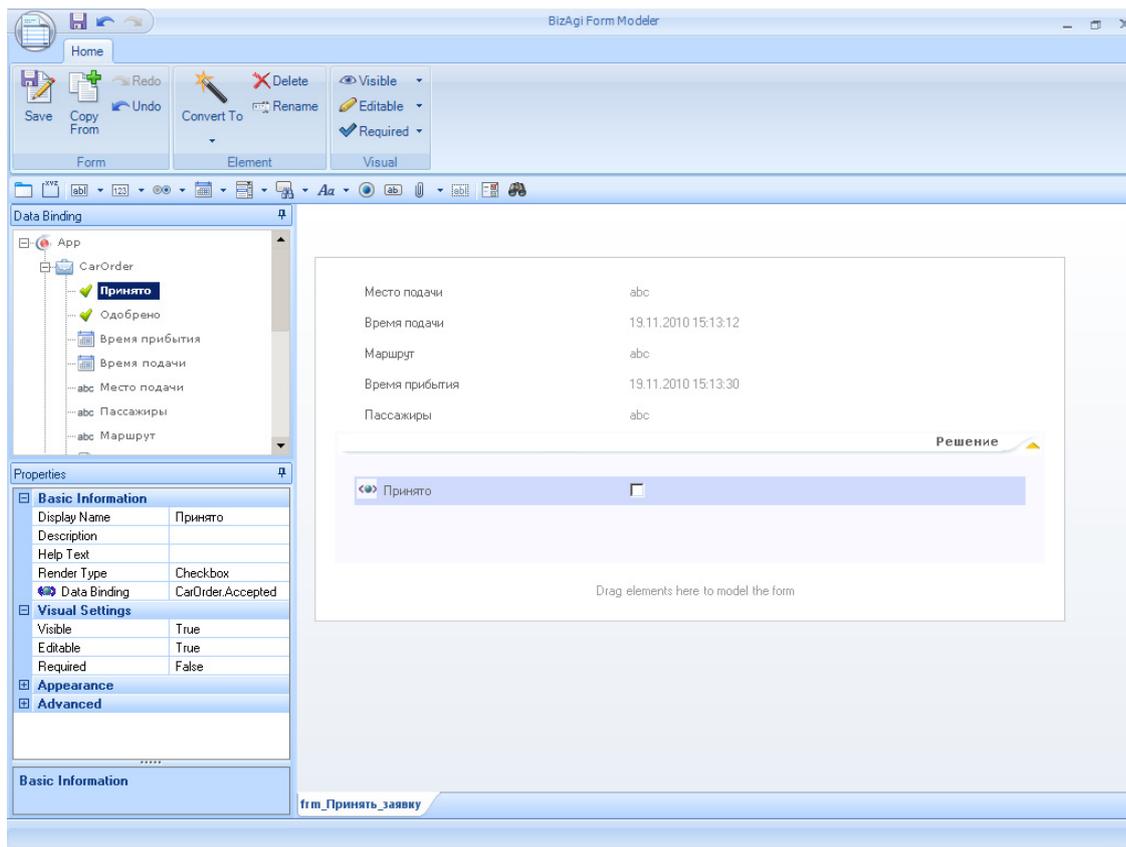
Сохраняем форму, переходим на форму информирования заказчика об отказе. Удаляем с формы атрибут «Одобрено»



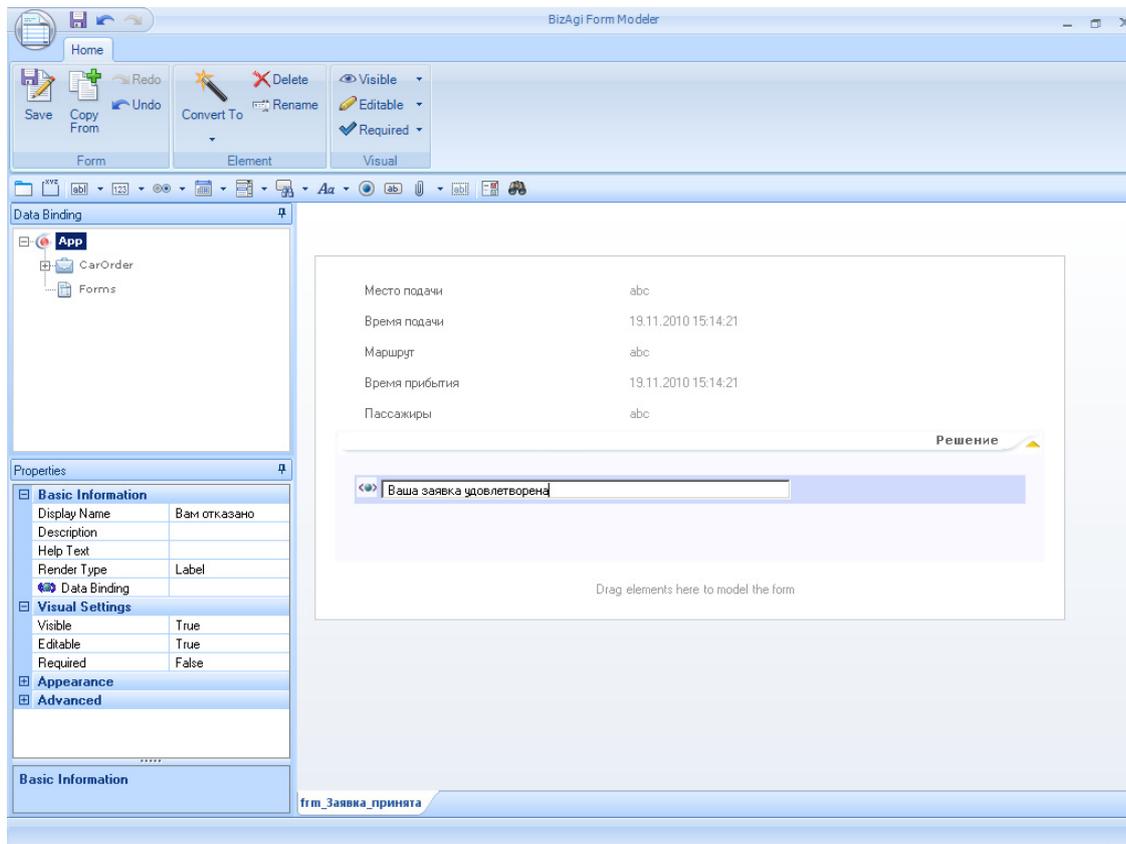
И добавляем статическую надпись:



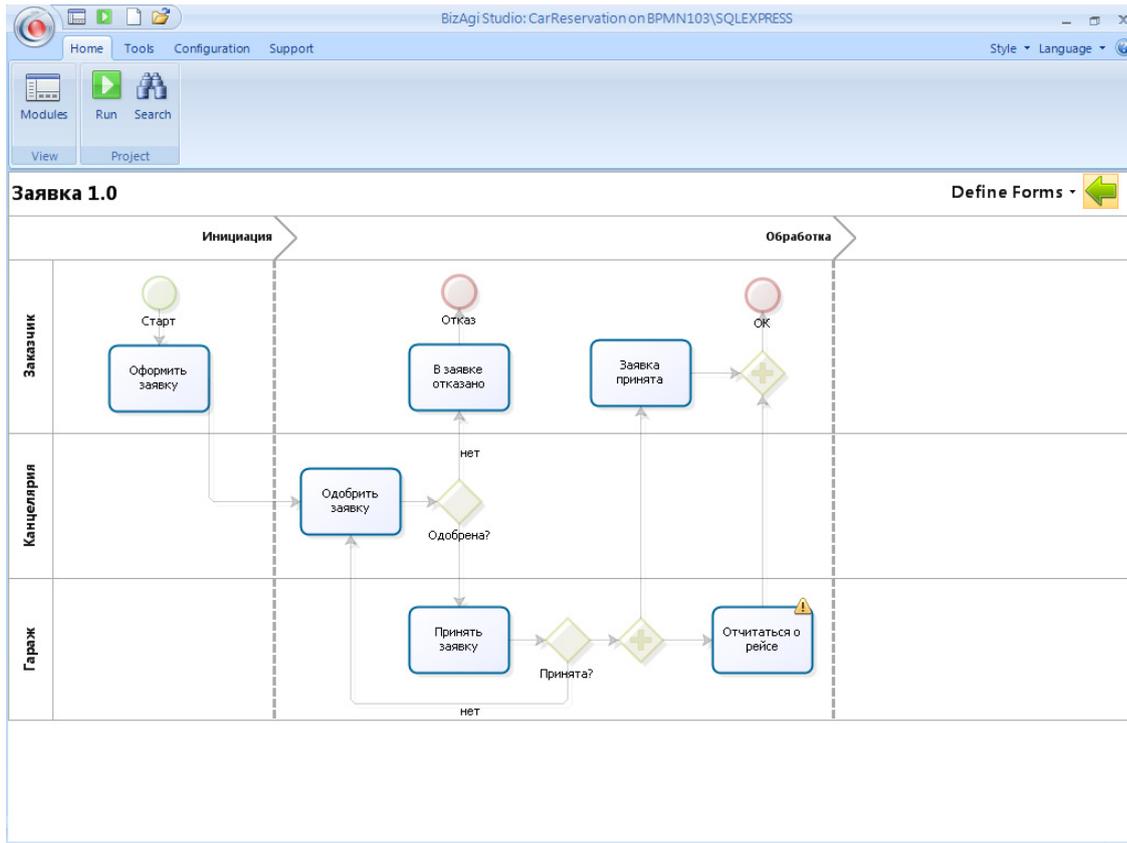
Аналогичным образом поступаем с формой приема заявки диспетчером гаража. Добавляем еще один булев атрибут «Принято».



Форма «Заявка принята» делается под копірку с «В заявке отказано», меняется только статическая надпись:



Мы определили все формы, кроме формы для шага «Отчитаться о рейсе». Ее пока отложим. Выходим из редактора форм, нажав стрелку. (Внимание: щелчок по крестику закрытия окна на этом экране закроет студию – для возврата в карусель пользуйтесь зеленой стрелкой.)

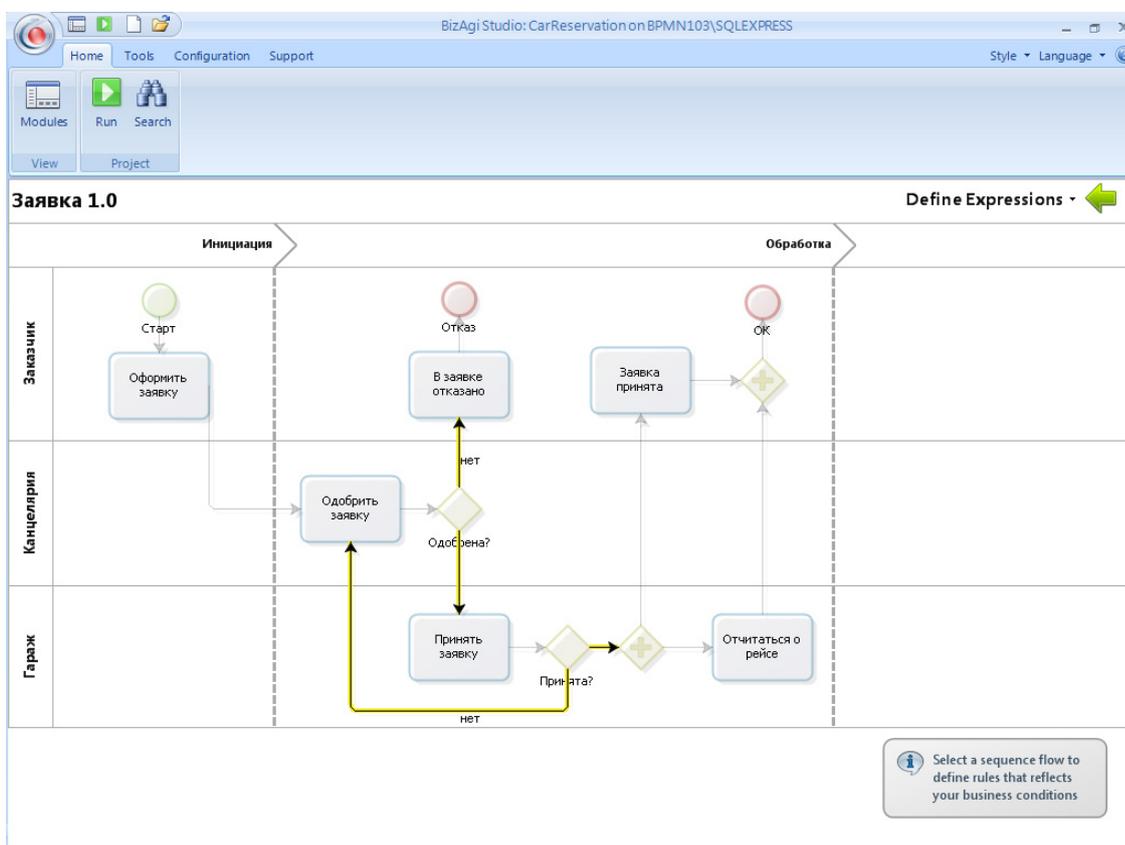


7. Правила переходов

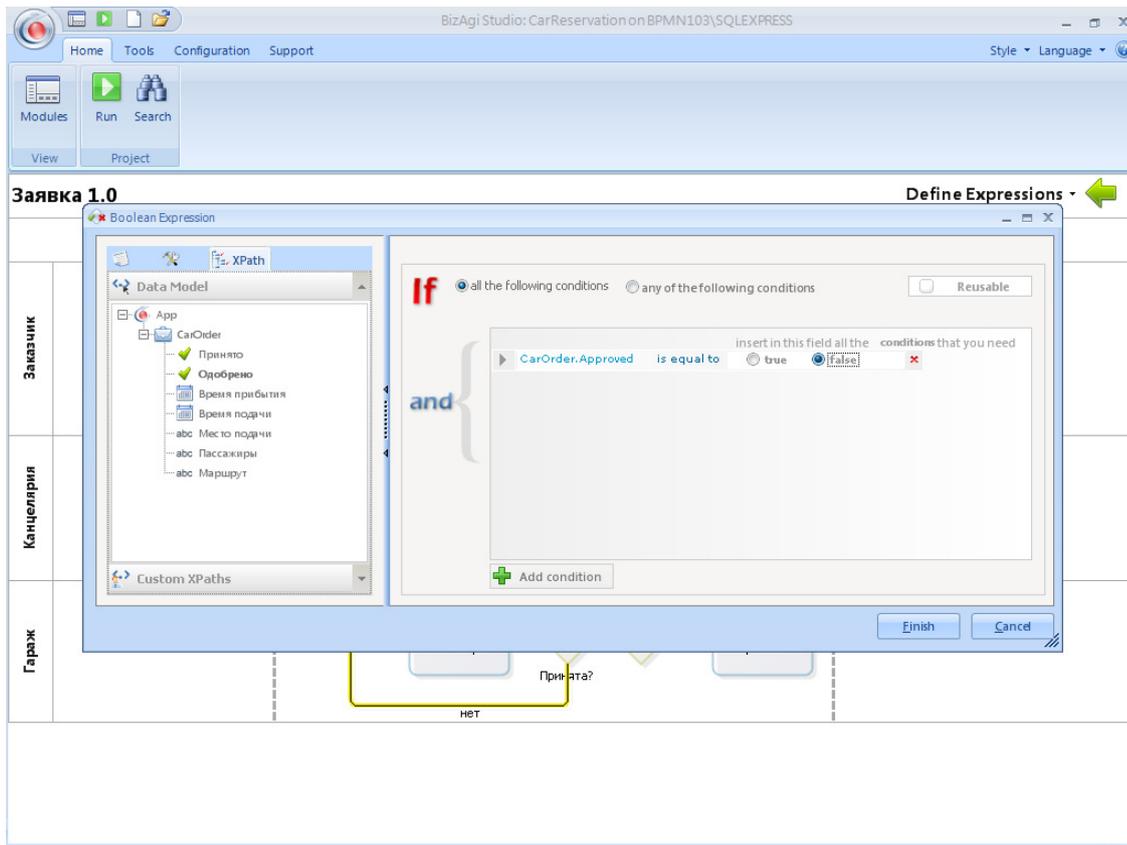
Теперь определим правила перехода для нашего процесса. Перейдите к шагу «Rules» и щелкните по надписи «Define Expressions».



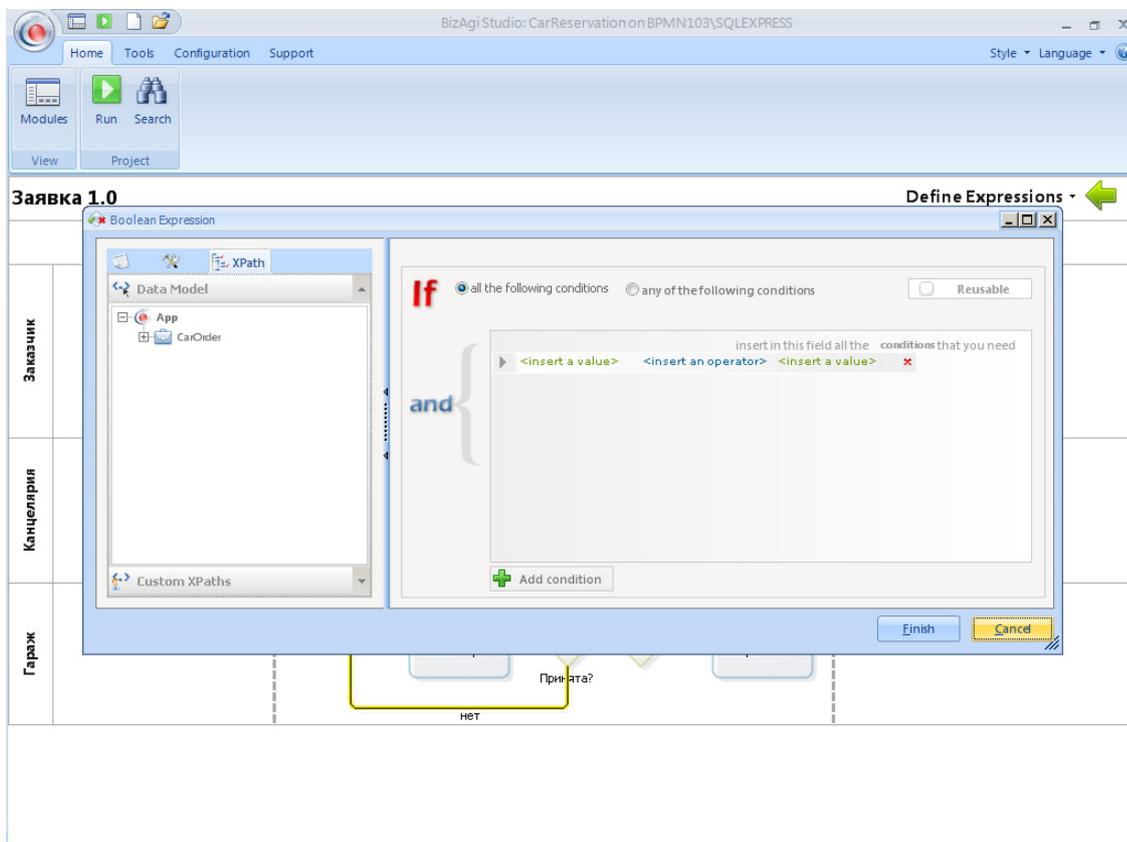
Желтым цветом выделены переходы, правила для которых не определены. Щелкните по стрелке, ведущей к задаче «В заявке отказано».



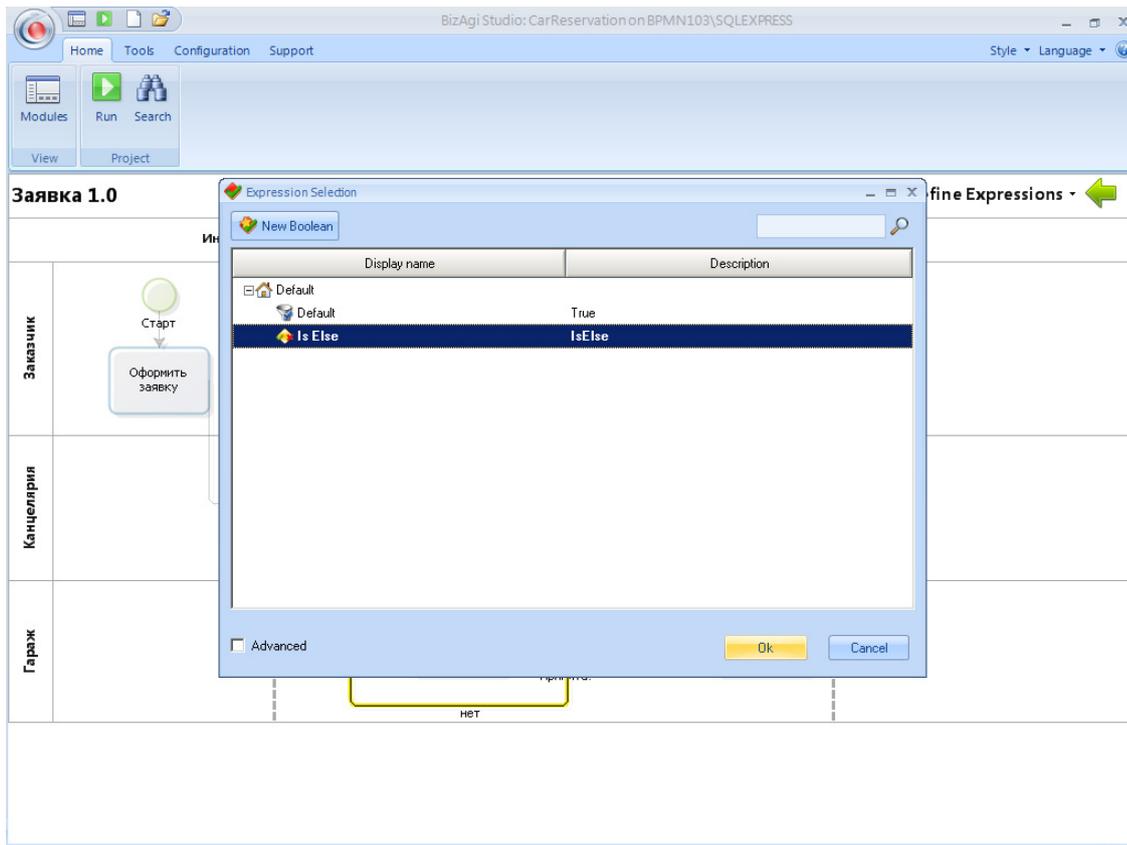
Условие перехода задается при помощи мыши. Перетащите атрибут «Approved» объекта «CarOrder», щелчком мыши выберите условие «is equal to» и значение «false». Условие определено, щелкните «Finish»:



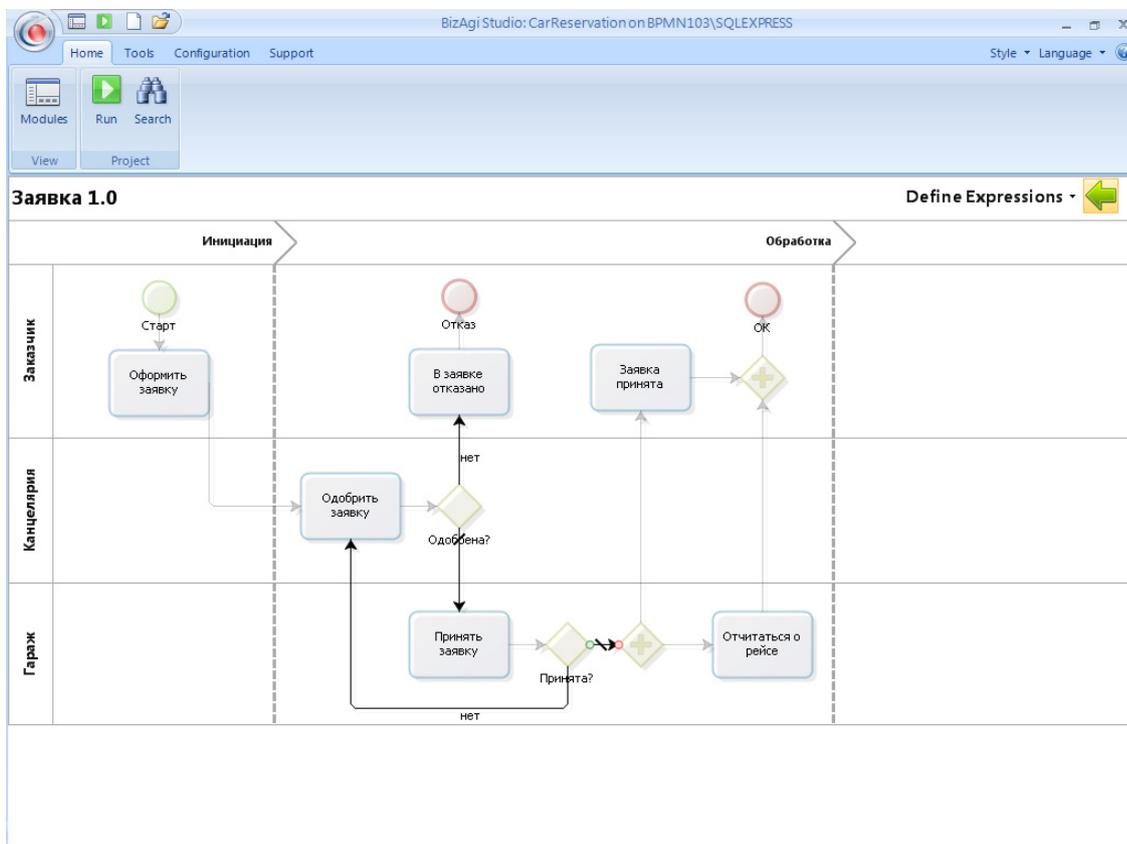
Теперь выберите стрелку, ведущую к задаче «Принять заявку». Для нее мы не будем задавать логическое условие, а сделаем ее переходом по умолчанию. Для этого на первой форме щелкните «Cancel»:



А на следующей выберите «IsElse». (Иконка с надписью «Default» тут сбивает с толку – для того, чтобы определить переход по умолчанию, надо выбирать не его, а «IsElse».) Щелкните «OK».



Аналогичным образом определите переходы для стрелок, выходящих из развилки «Принята». Завершив работу, щелкните зеленую стрелку, чтобы вернуться в карусель.

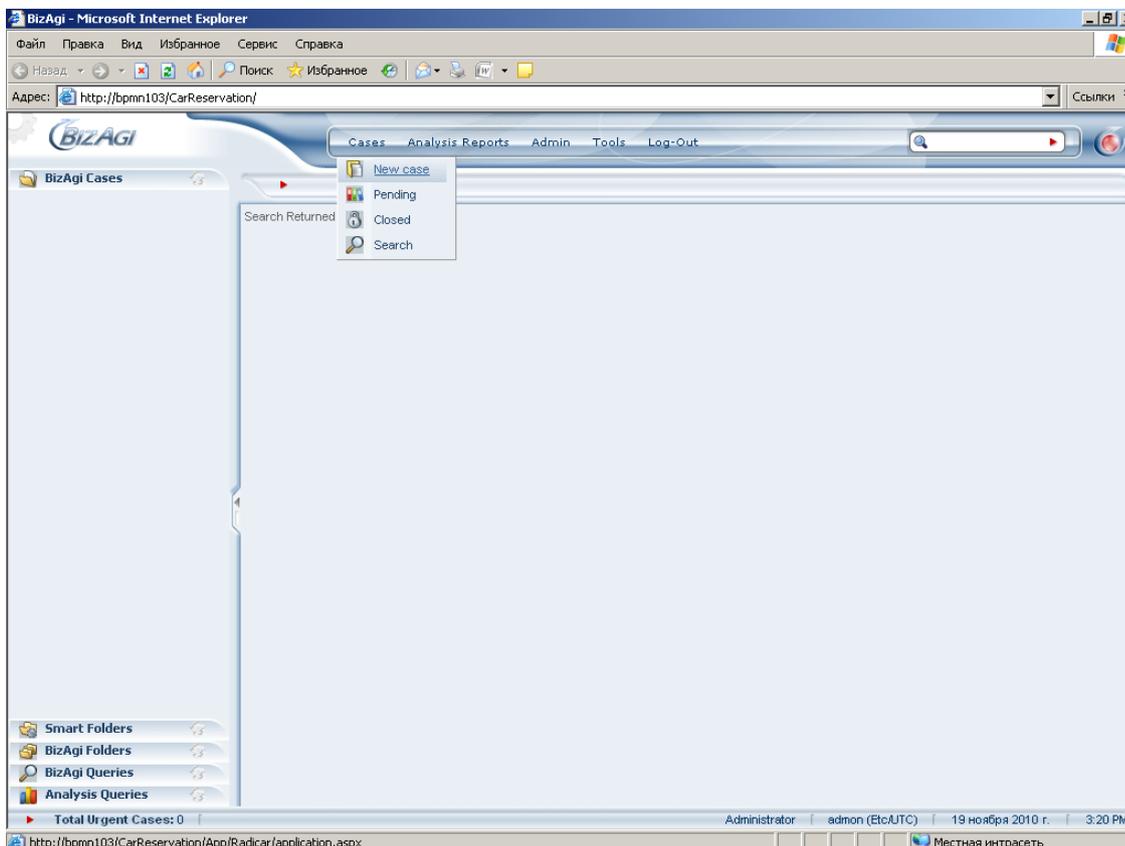


8. Исполнение

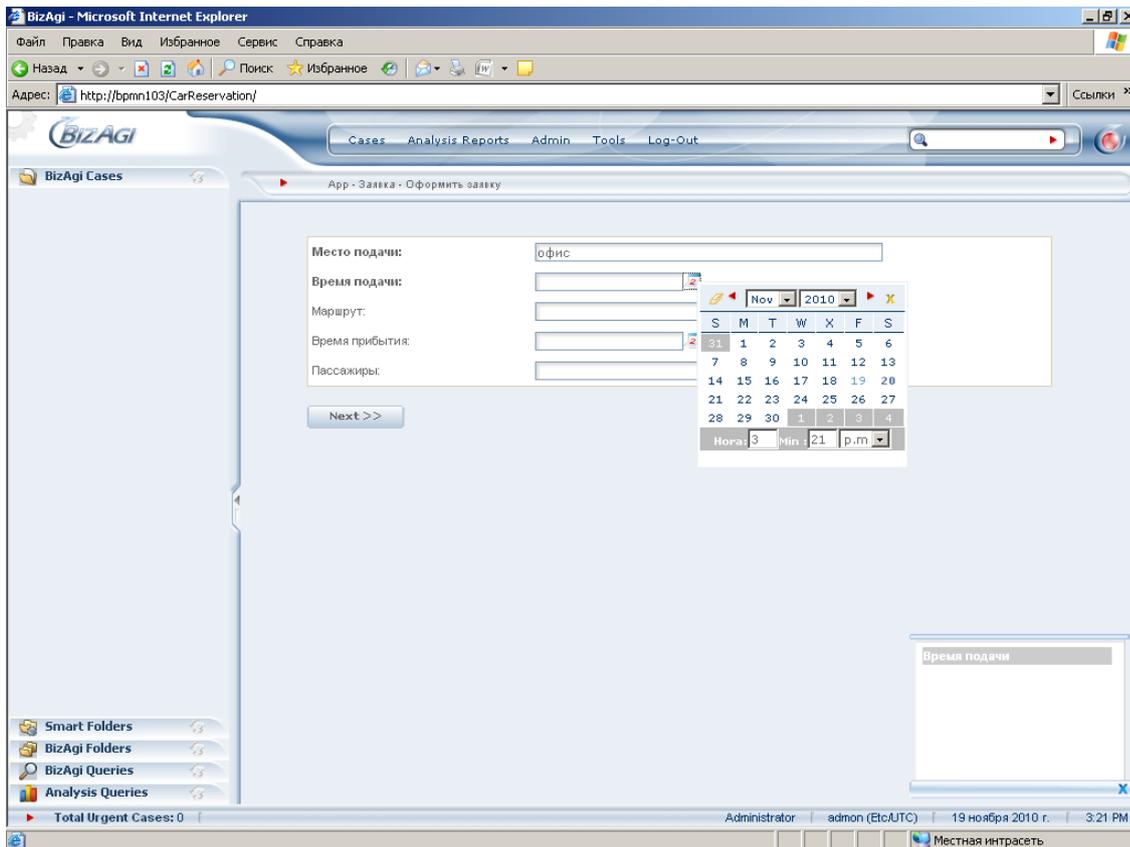
Процесс можно запустить на исполнение. Для этого щелкните по кнопке «Run»:



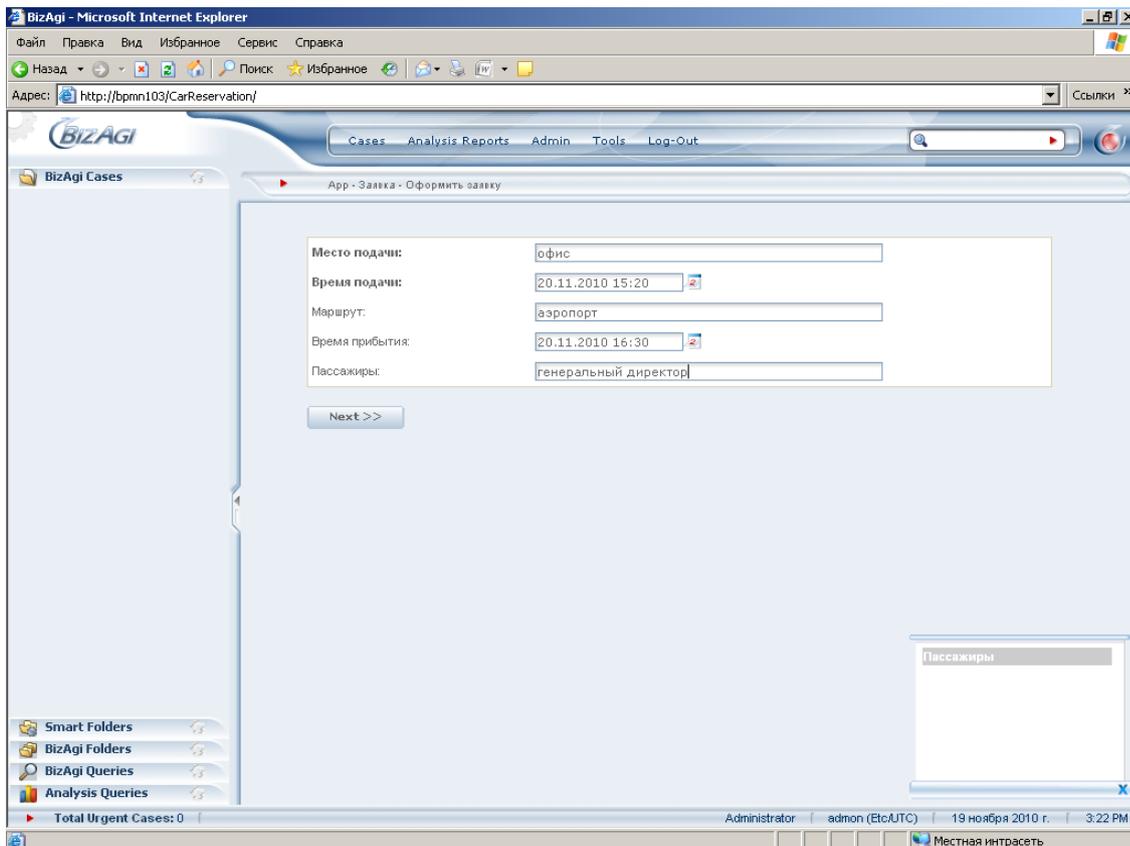
Запустится браузер, и через некоторое время в нем откроется портал «BizAgi Xpress». Логика пользовательского интерфейса портала такова, что там, где нет выбора, он пользователю и не предлагается. В нашем проекте пока что определен всего один процесс, и пользователь тоже только один – администратор. Поэтому портал не запрашивает имя-пароль при входе, а когда мы стартуем процесс («Cases» - «New case»)



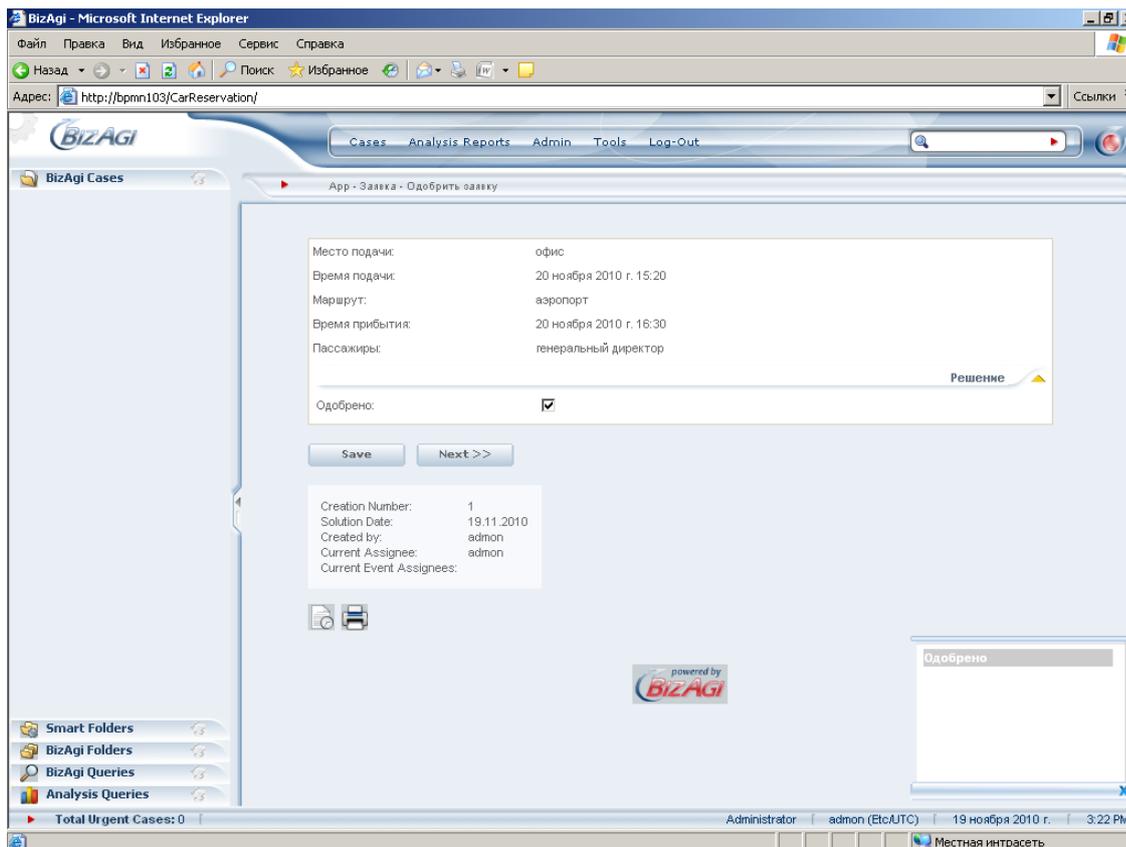
то без лишних вопросов запускается процесс «Заявка»:



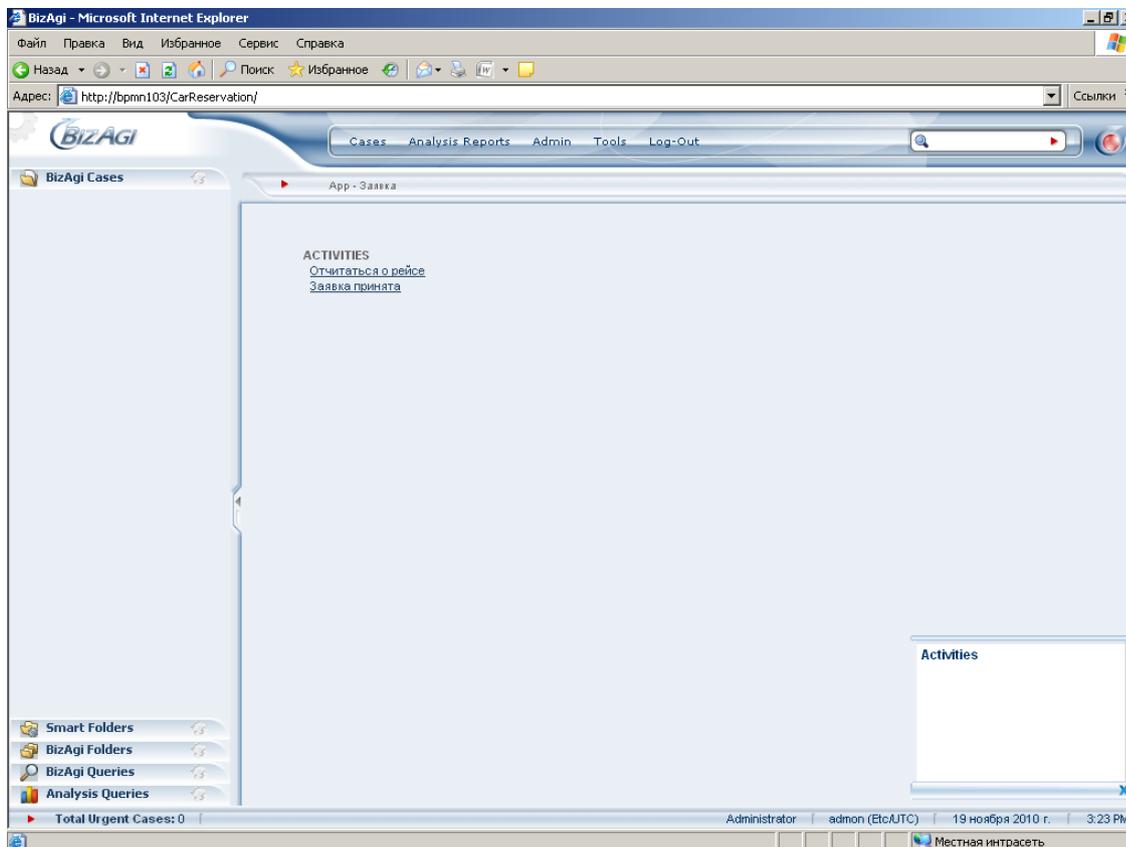
Оформляем заявку, щелкаем «Next»:



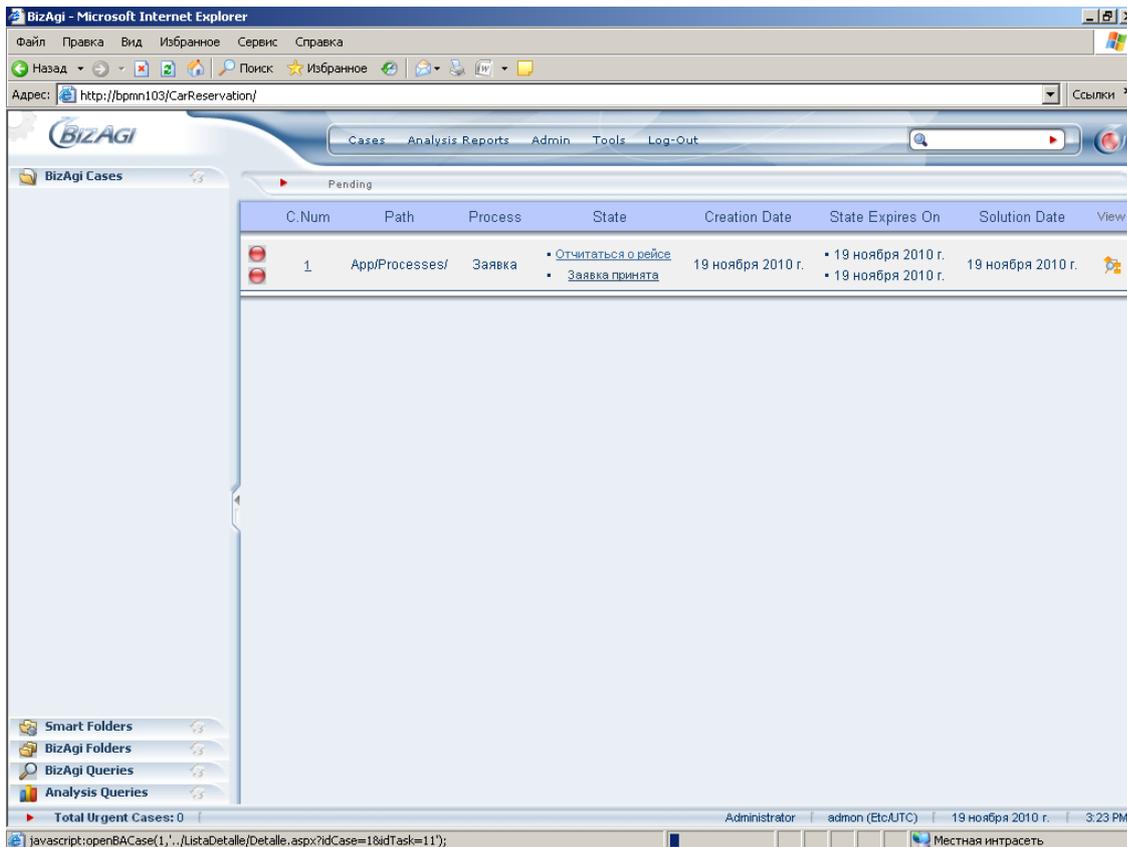
Поскольку исполнитель для следующей задачи «Одобрить заявку» остался тем же (у нас всего один пользователь), то портал автоматически предлагает выполнить и ее тоже. Ставим галочку в поле «Одобрено», щелкаем «Next»:



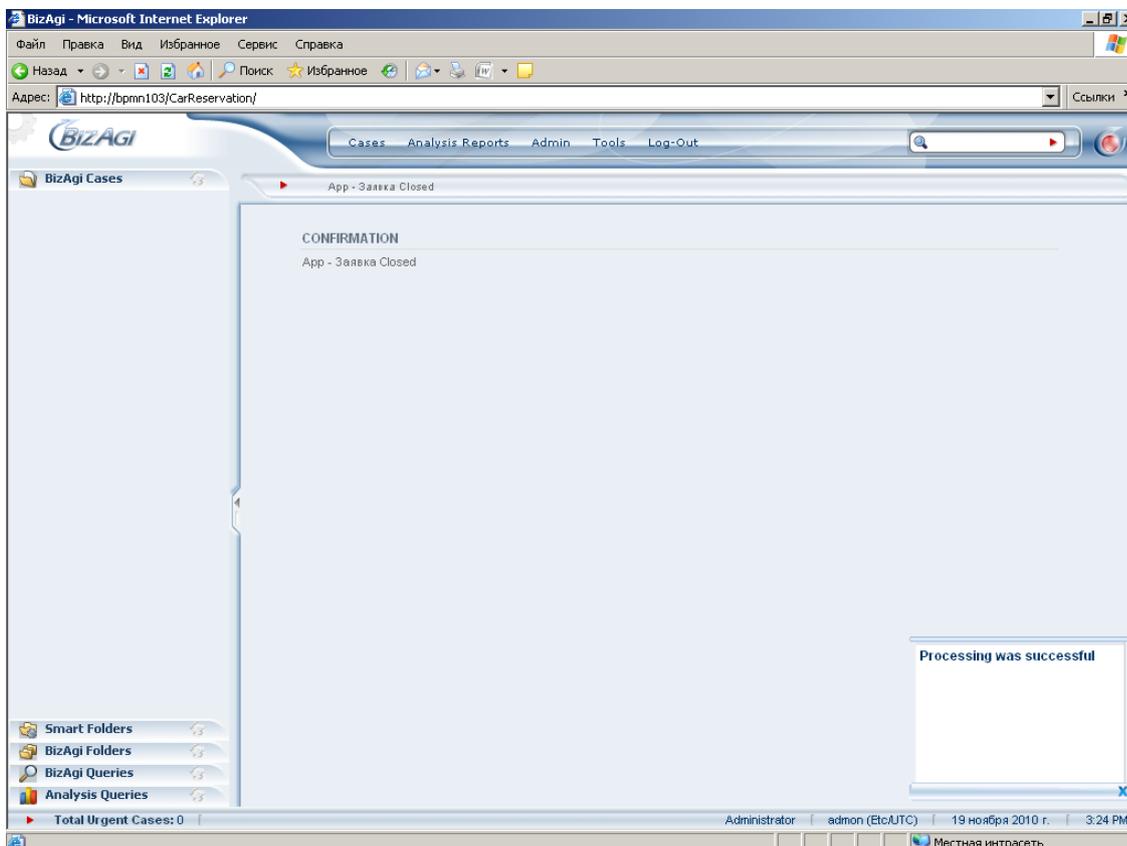
Аналогичным образом проходим следующий шаг «Принять заявку». Поскольку следом по схеме процесса идет параллельная развилка, портал предлагает на выбор две задачи:



Задачи, ожидающие выполнения, в любое время можно увидеть, щелкнув «Cases» - «Pending»:

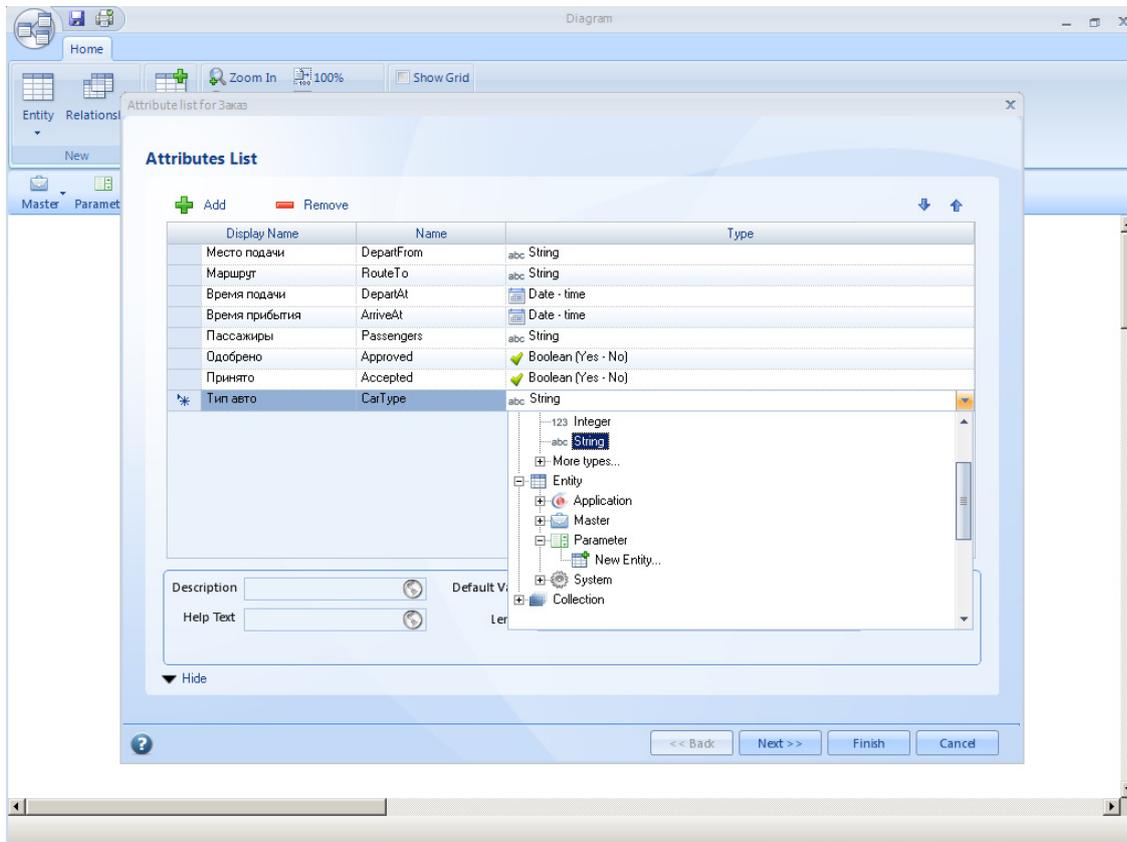


Выполним обе задачи (в любой последовательности) и тем самым завершим процесс. Поскольку для задачи «Отчитаться о рейсе» форму мы пока не определили, просто щелкните «Next». На экране появится уведомление о завершении процесса:

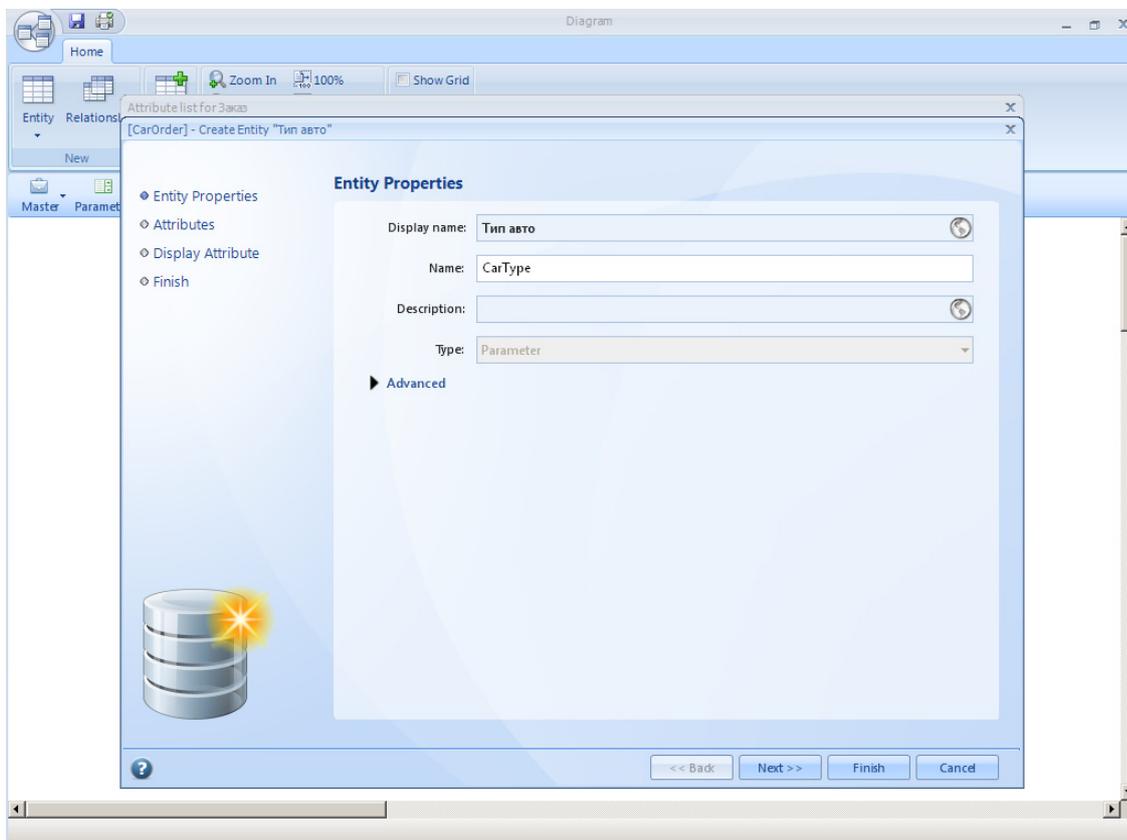


9. Справочники

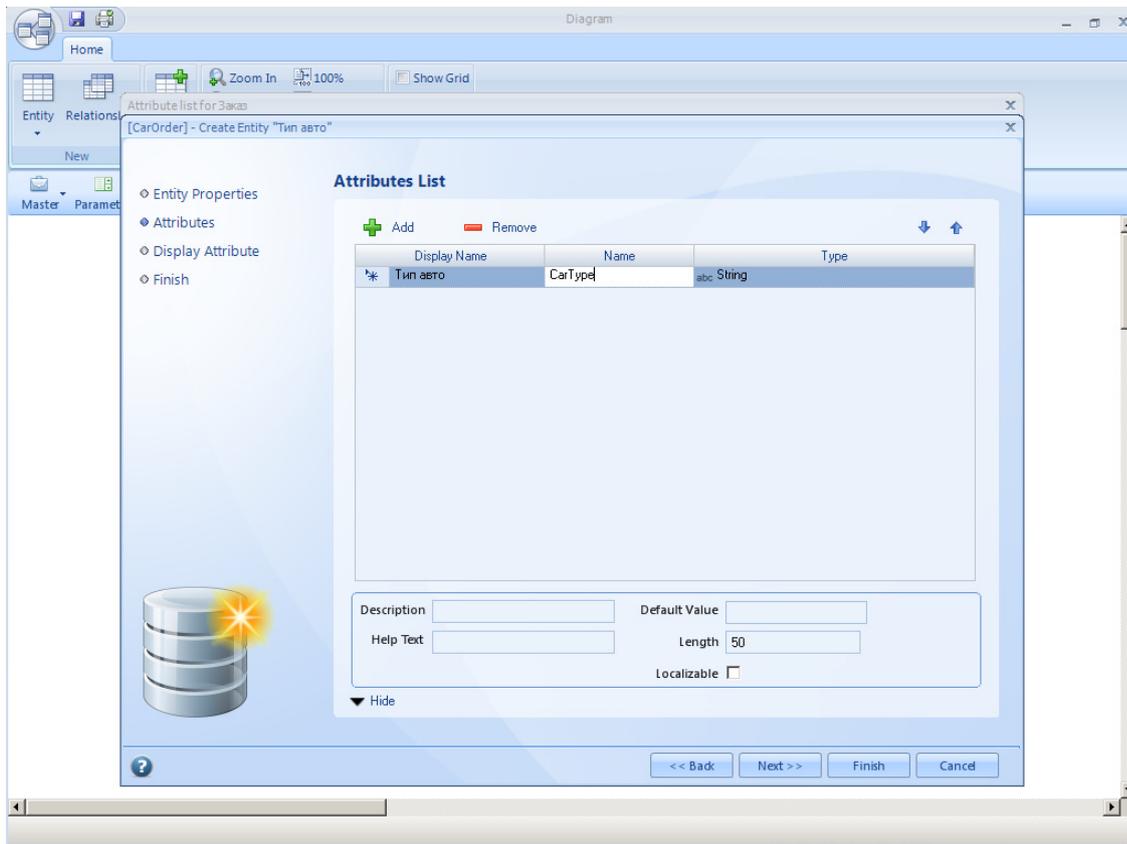
Помимо обычных атрибутов – строк, чисел, дат – любая компьютерная система оперирует справочниками. Добавим в нашу систему справочники типов автомобилей (легковая, грузовая и т.д.) и самих автомобилей, имеющих в нашем гараже. Заходим в моделирование данных и добавляем атрибут. В качестве типа атрибута выбираем «Entity» - «Parameter» - «New Entity»:



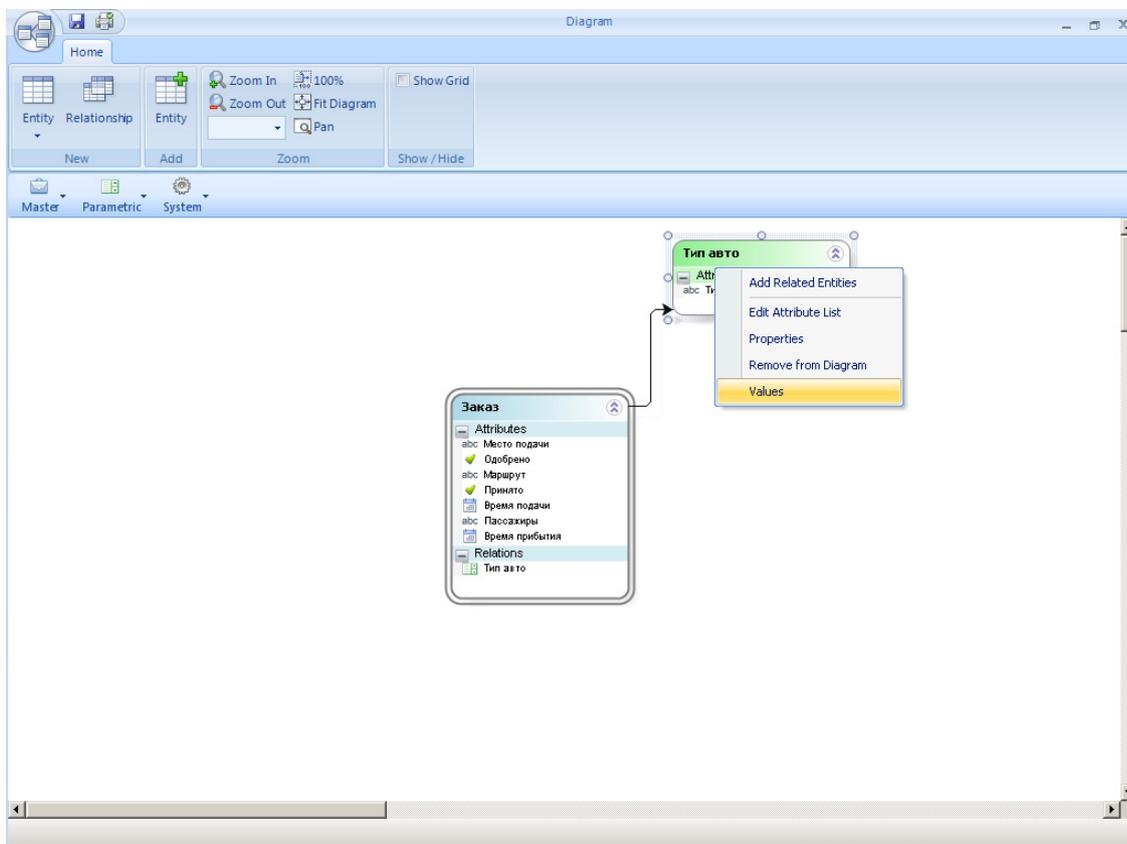
Создаем справочник типов автомобилей:



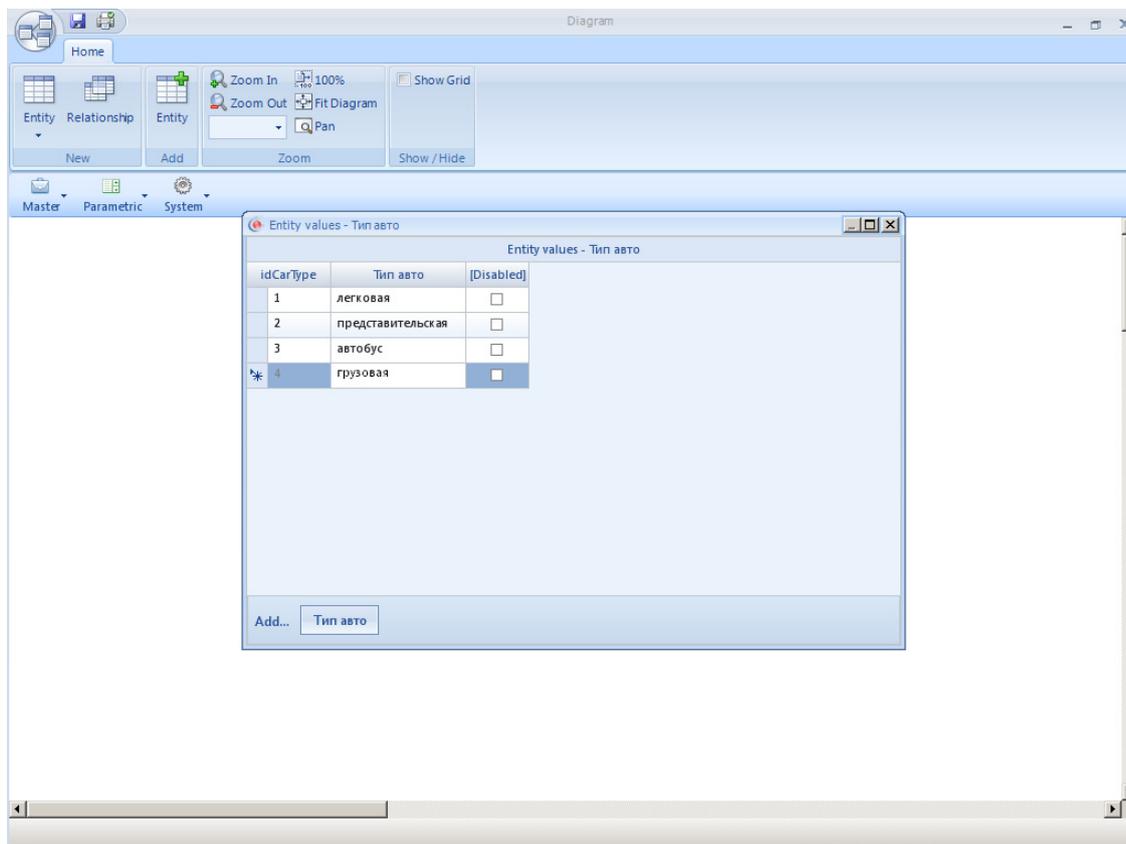
В справочнике будет всего один атрибут: собственно тип автомобиля:



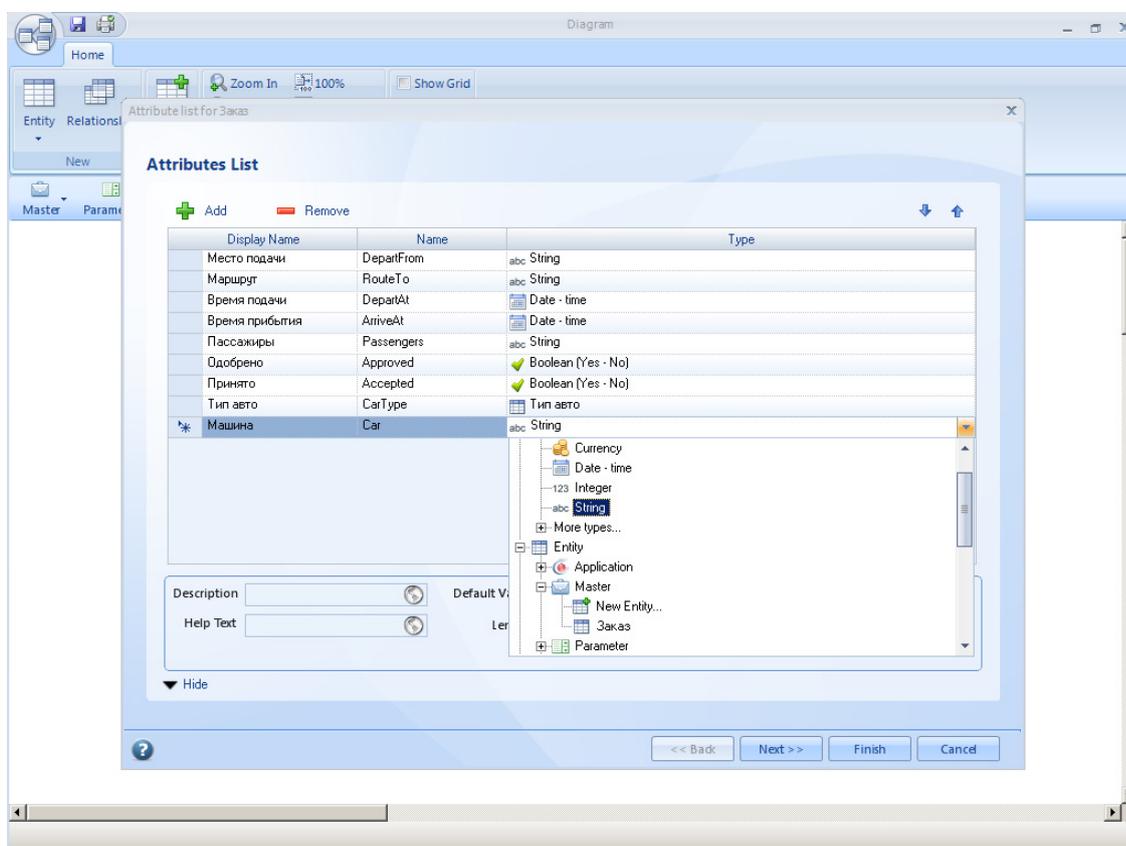
Кликом по «Finish» заканчиваем сначала редактирование справочника, затем редактирование объекта «Заказ». На диаграмме появился созданный нами справочник. Заполним его значениями: правая кнопка – «Values»:



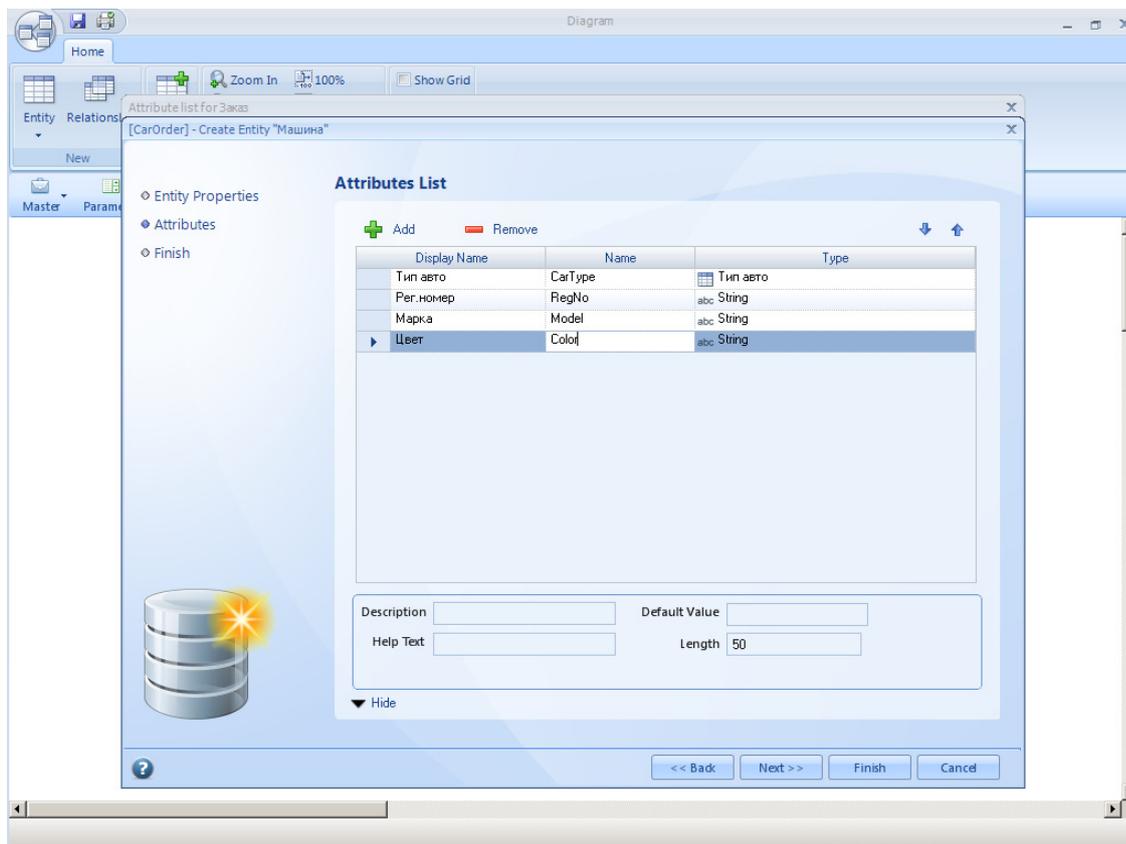
Щелкая по кнопке «Тип авто», создаем новые записи.



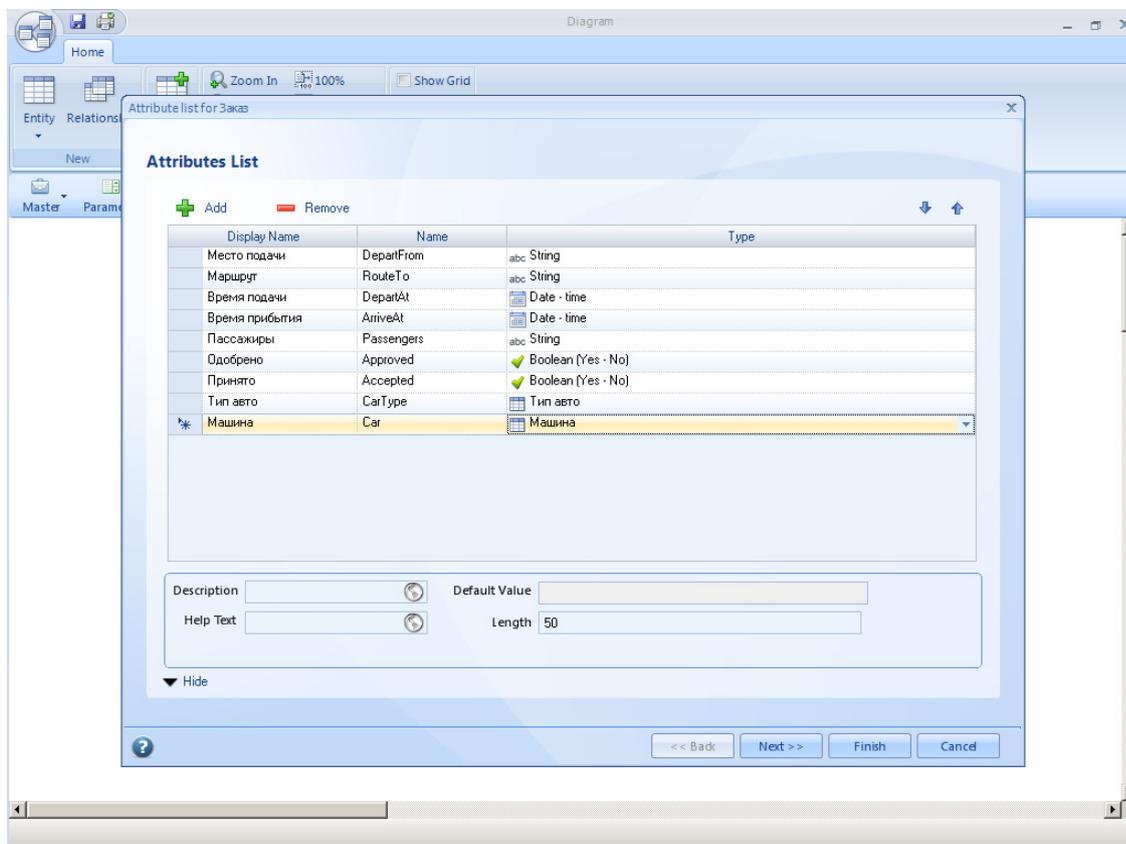
Теперь создадим справочник автомобилей. Делаем это аналогично справочнику типов автомобилей, только вместо «Entity» - «Parameter Entity» выбираем «Entity» - «Master Entity». Общее правило тут следующее: если какие-то данные должны быть в системе, даже когда в ней нет ни одного экземпляра процесса, то это Parameter Entity. Если же данные добавляются, правятся, удаляются по мере работы, то это Master Entity.



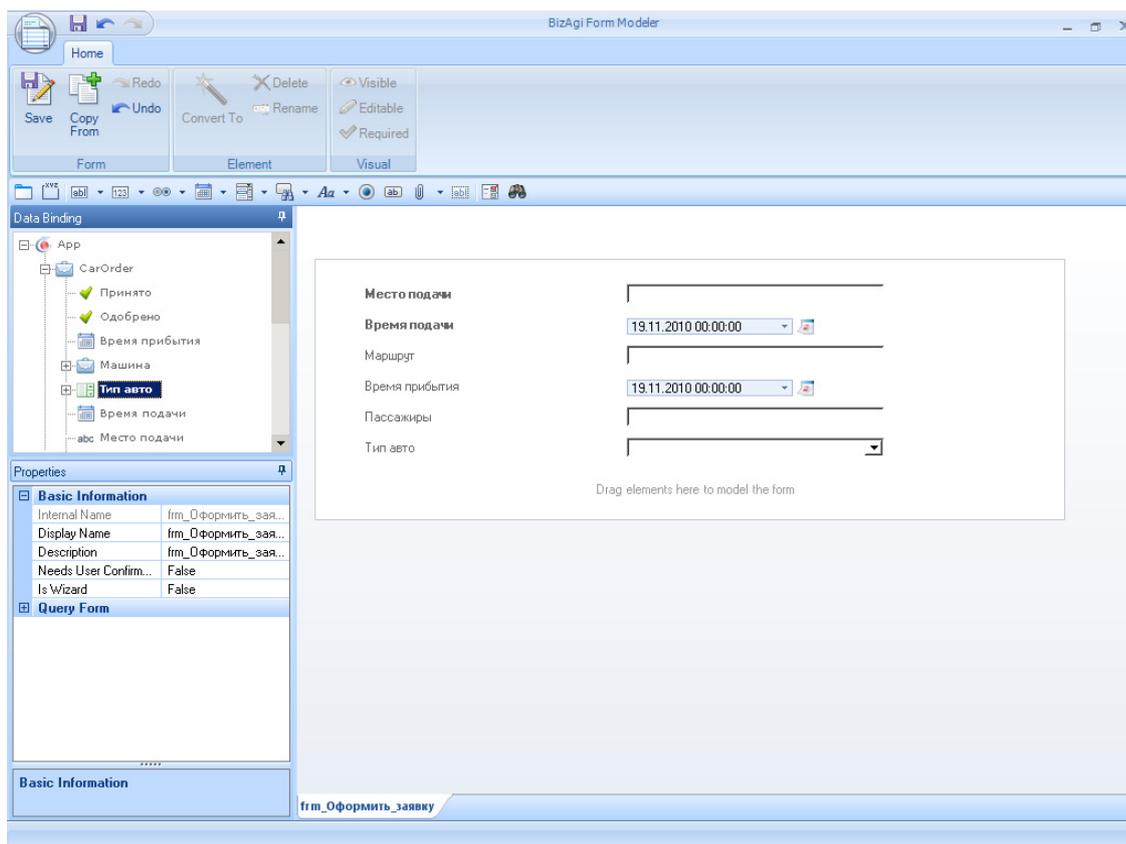
Вводим атрибуты базы данных автомобилей, и среди них тип автомобиля.



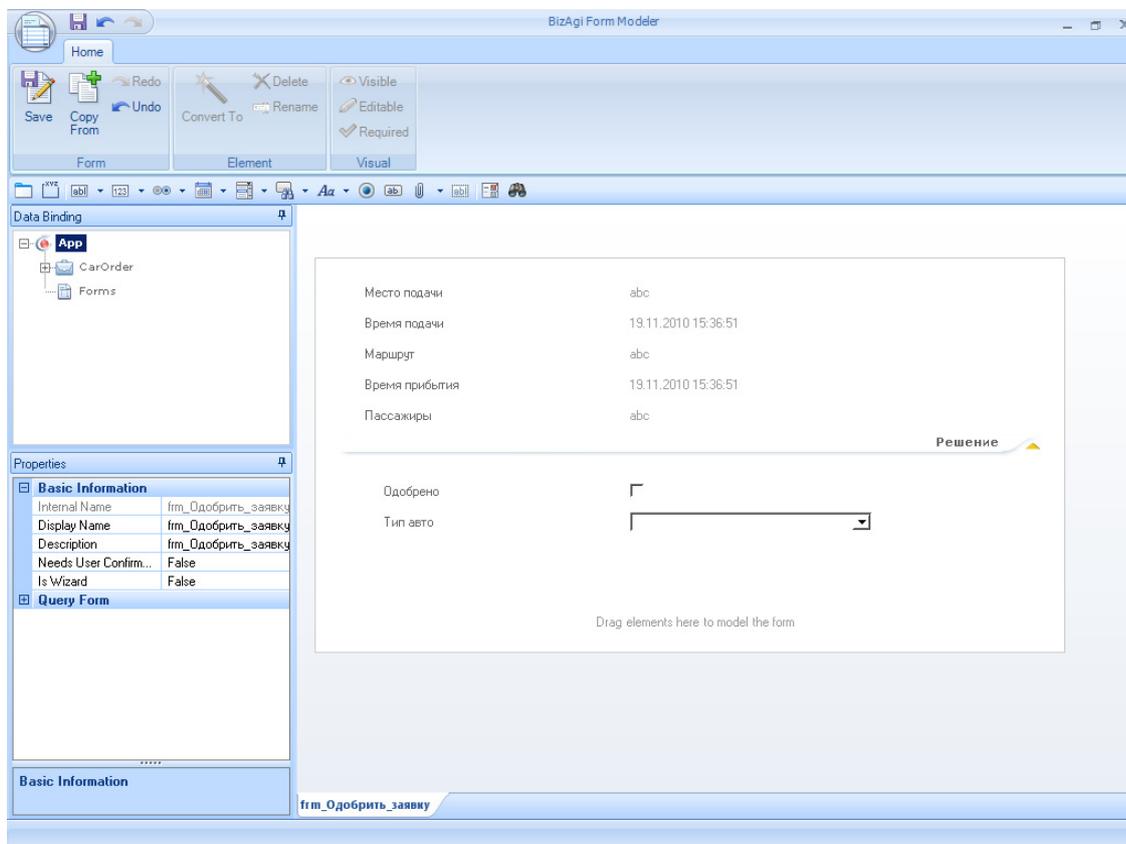
Теперь заказ содержит ссылку на тип автомобиля – это то, что пожелал заказчик, и автомобиль – то, что назначил заказу диспетчер гаража.



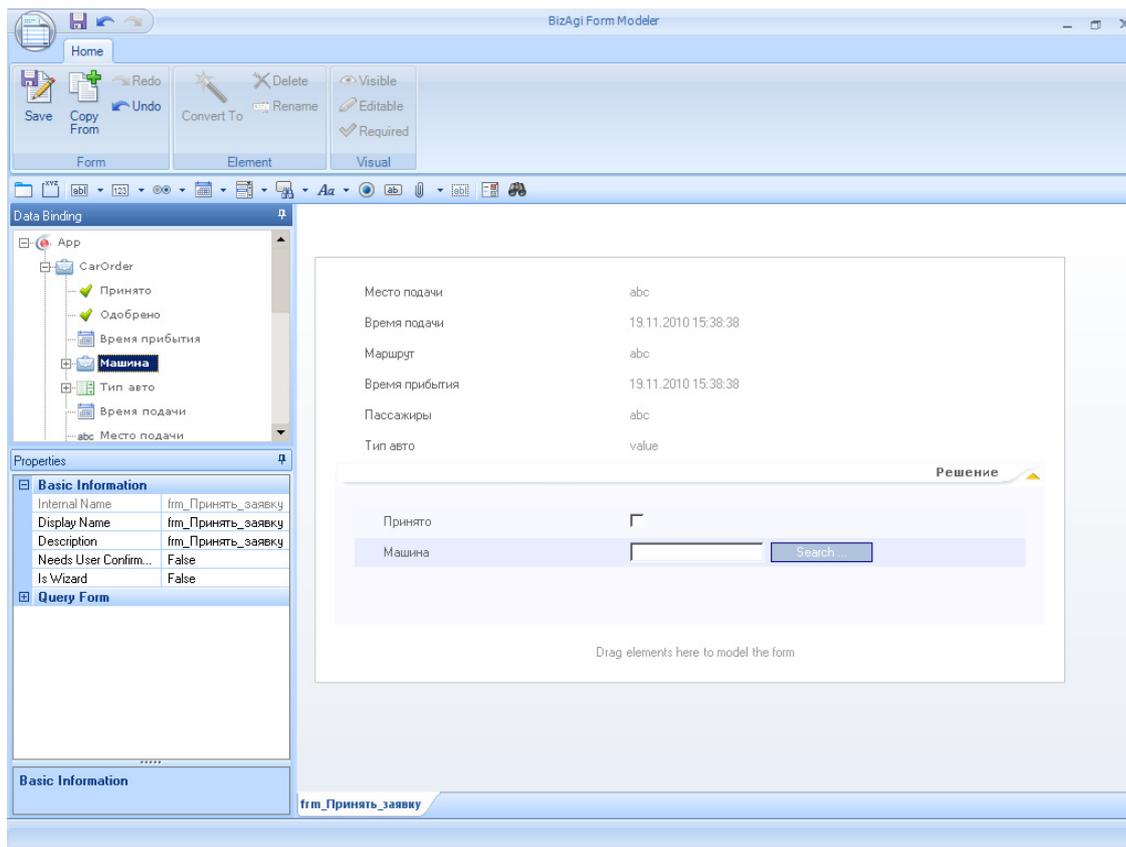
Добавляем тип авто на форму создания заявки. Для этого просто перетаскиваем мышью объект «Тип авто», не раскрывая его, и форма дополнится выпадающим списком:



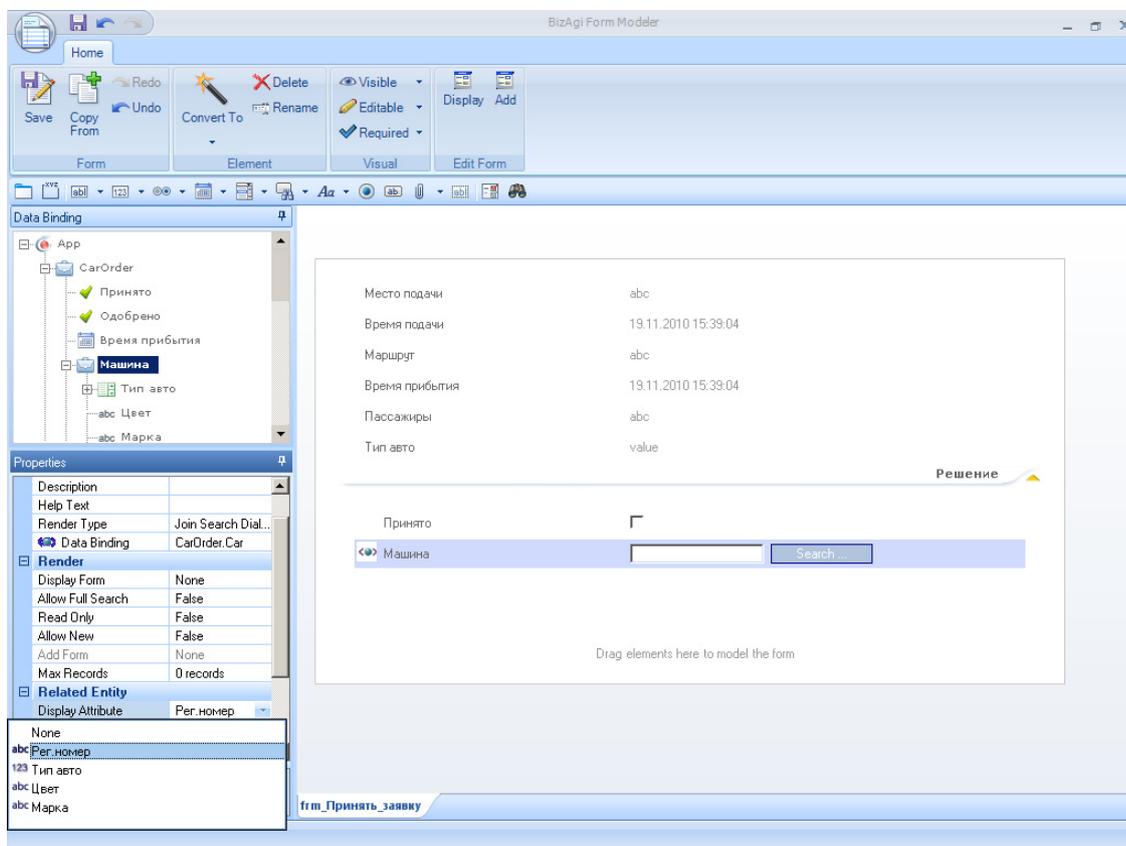
Дадим секретарю возможность поменять тип автомобиля на форме задачи «Одобрить заявку»:



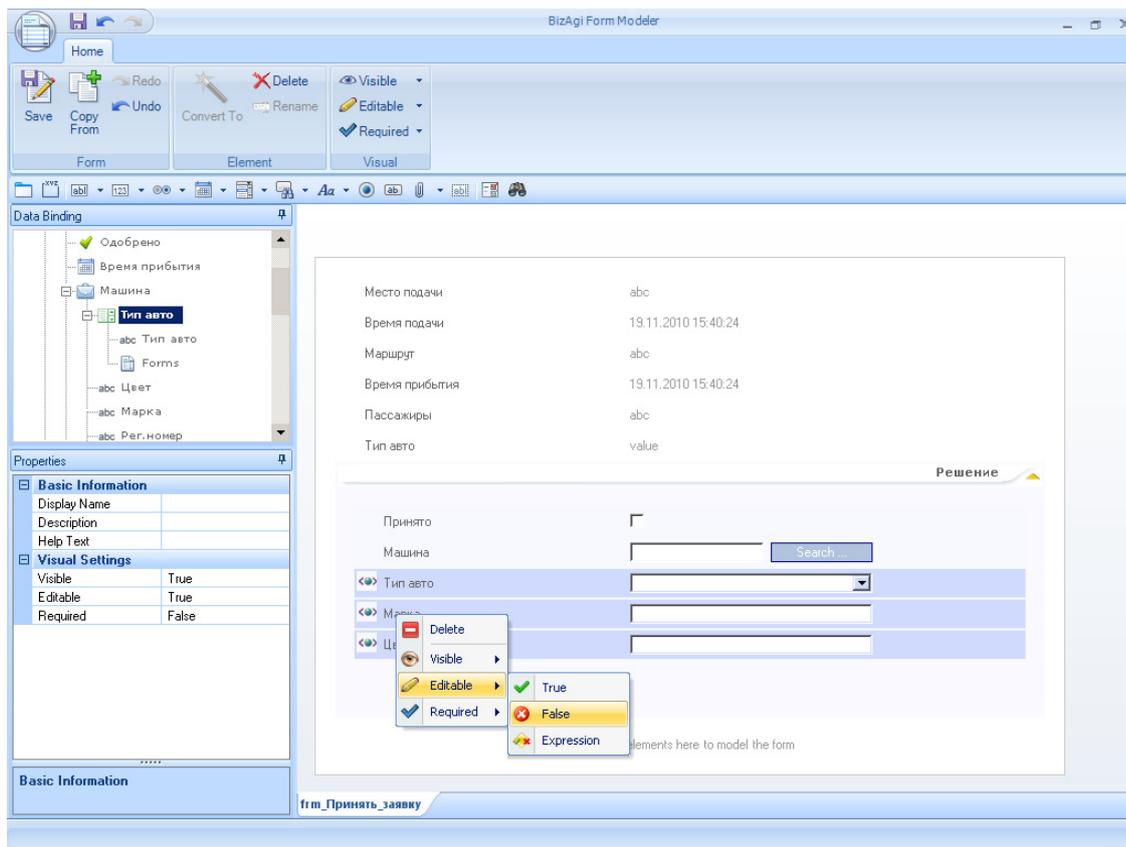
Логика работы со справочником автомобилей несколько сложнее: тут надо иметь возможность искать по потенциально большому справочнику. Также мы разрешим расширять справочник «на лету». Открываем форму задачи «Принять заявку» и перетаскиваем на форму справочник «Машина», не раскрывая его:



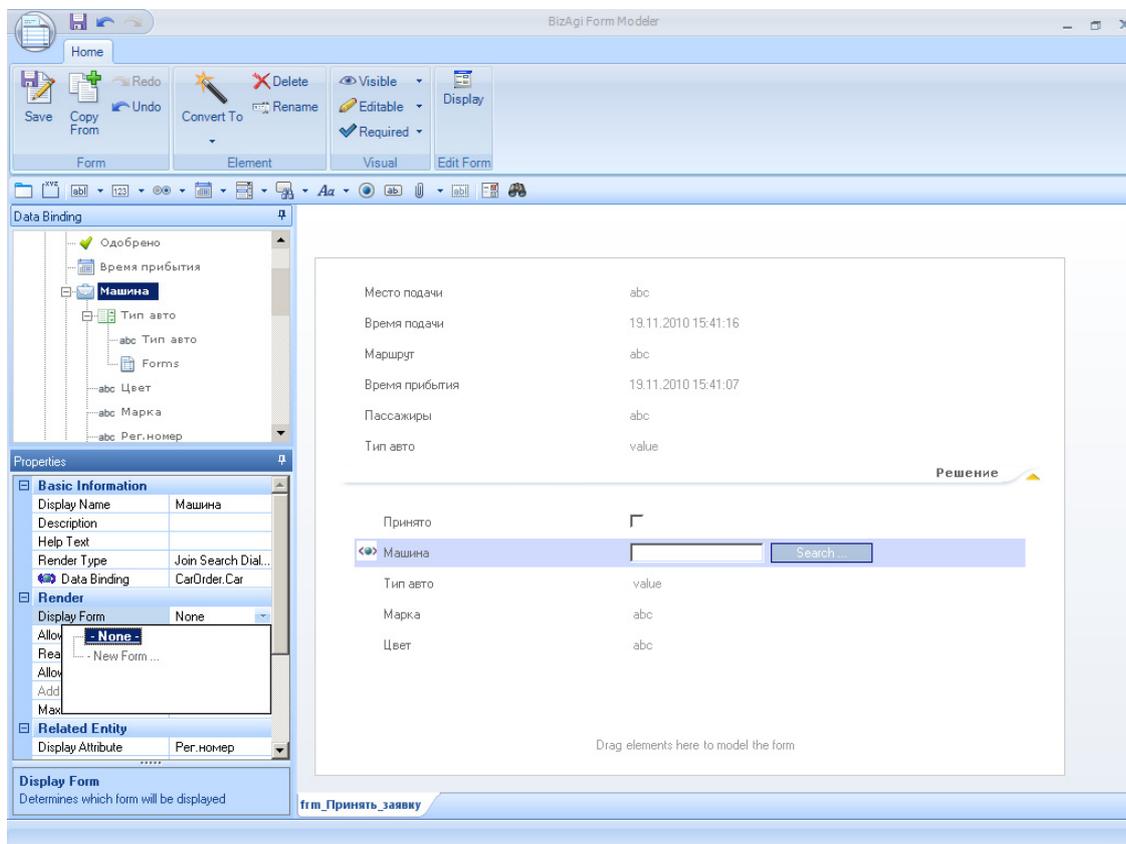
Теперь нам надо указать какой атрибут будет основным при поиске и задании машины. Очевидно, наиболее подходящий атрибут – регистрационный номер. Задаем его в качестве «Display Attribute»:



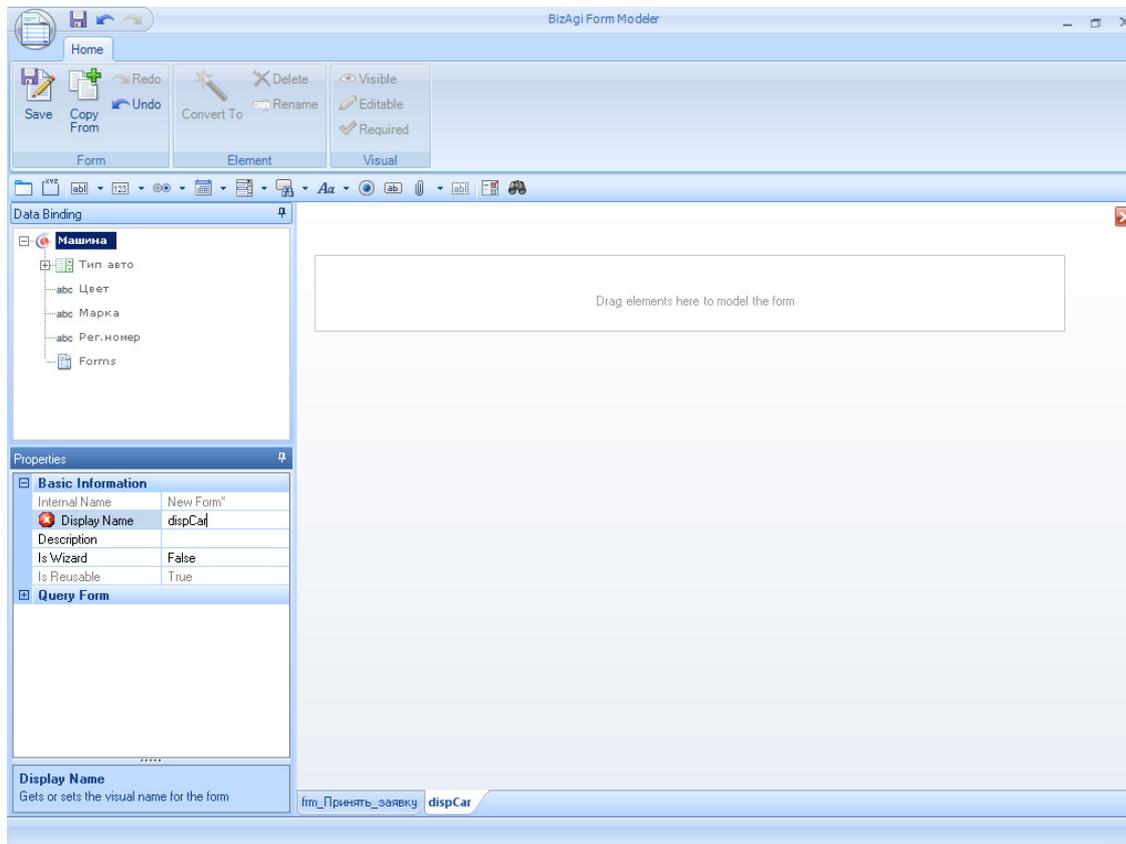
Как только машина выбрана, можно извлечь из таблицы и показать на форме и остальные атрибуты: марка, цвет, тип автомобиля. Добавляем эти атрибуты и запрещаем их редактирование:



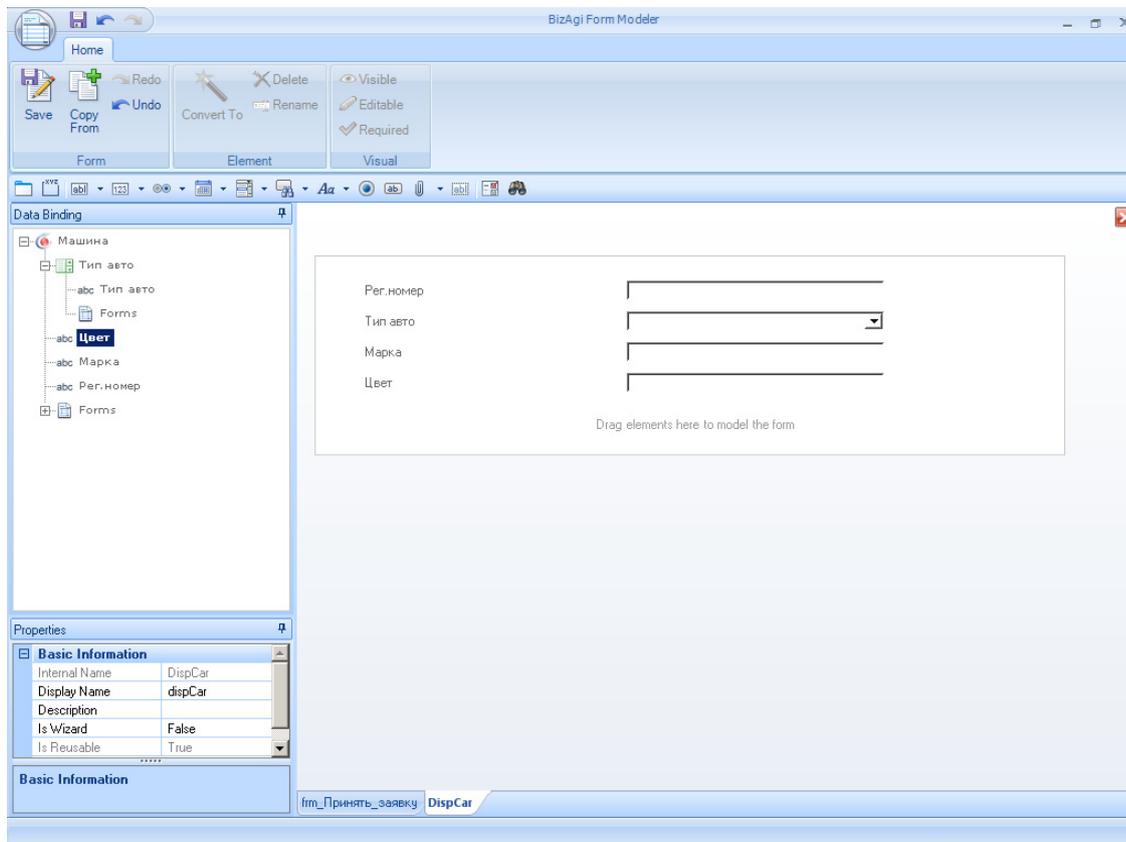
Теперь надо определить вспомогательные формы для поиска и для ввода новых автомобилей. Из выпадающего списка для «Display Form» выбираем «New Form».



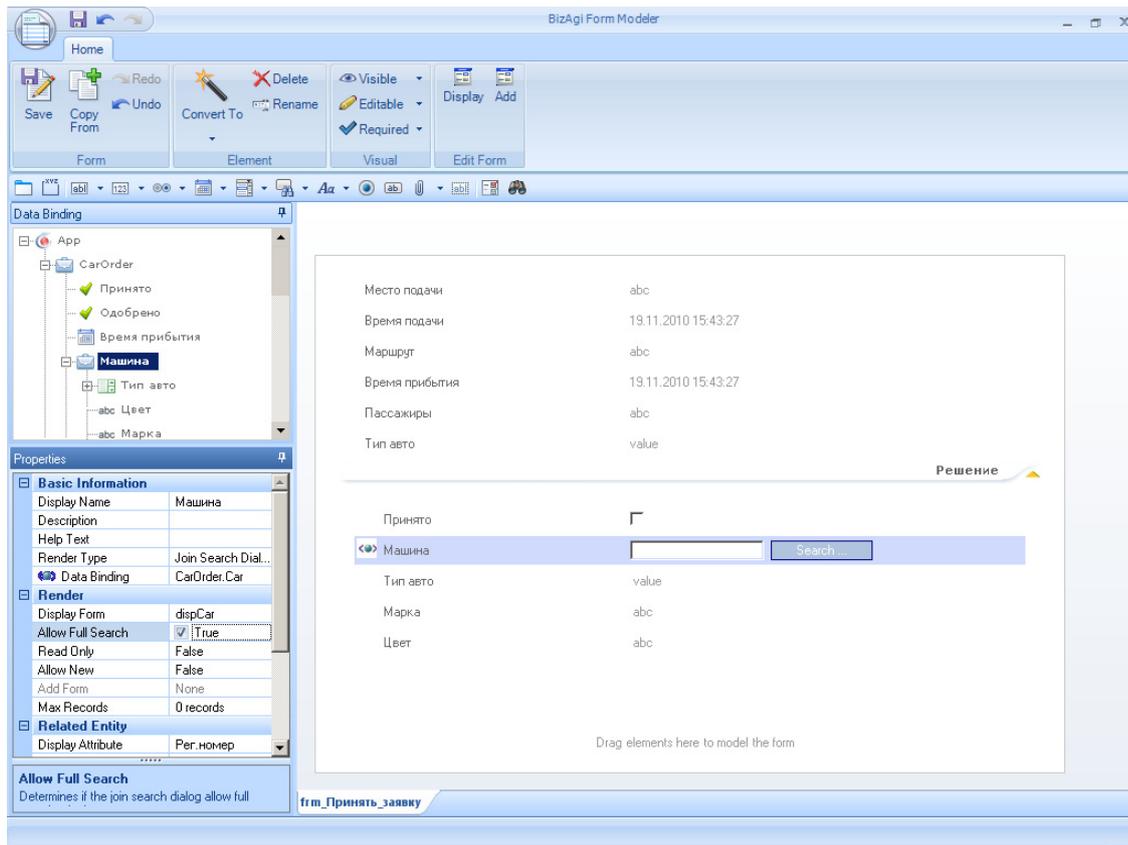
Открывается новая форма, пустая.



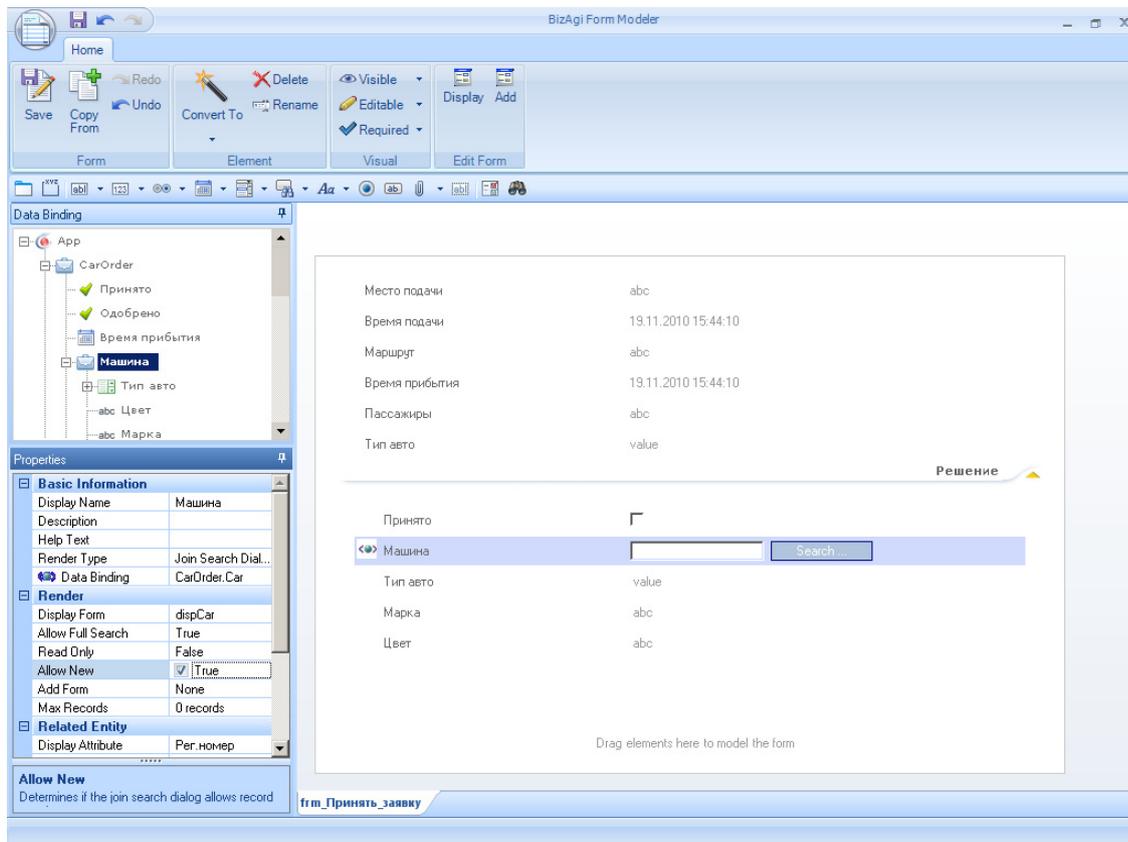
Задаем имя этой формы «dispCar» и набрасываем на форму атрибуты, по которым будем искать автомобили. Для выхода щелкаем по красному крестику в правом верхнем углу рабочего поля формы:



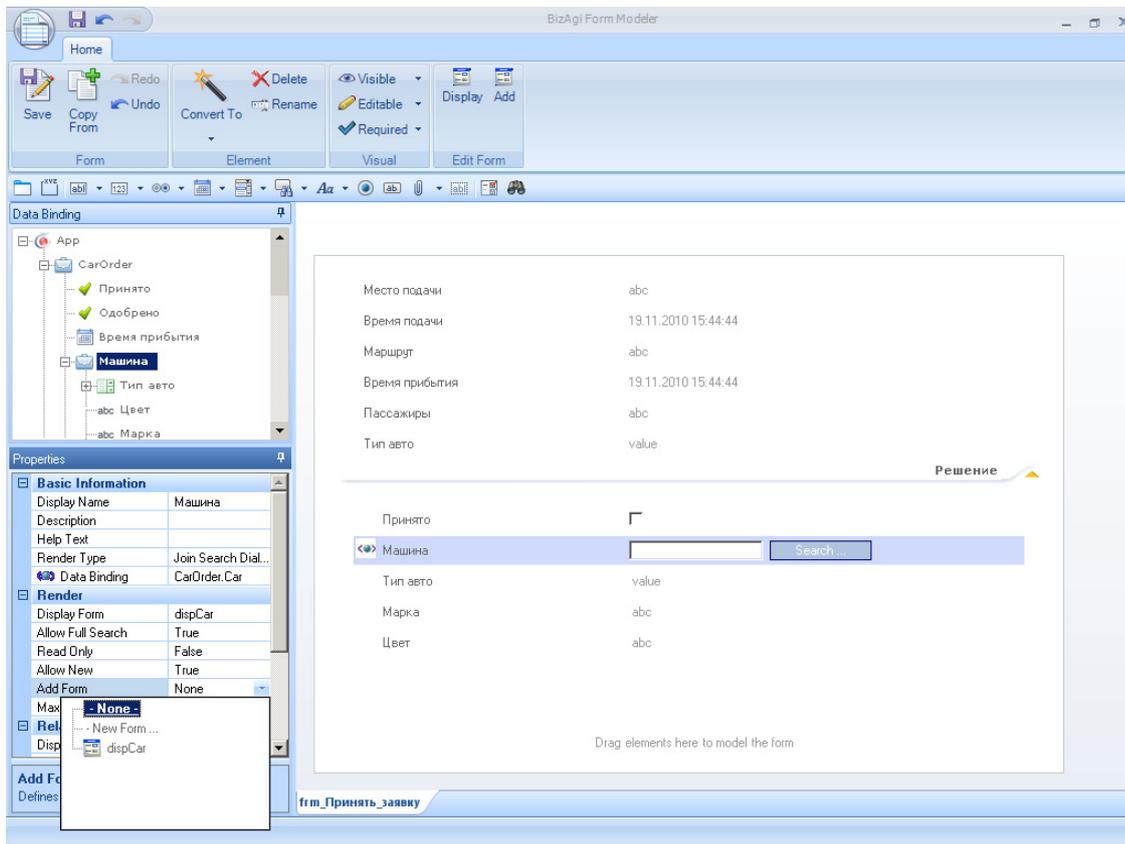
Закрываем форму поиска с сохранением, возвращаясь на форму задачи. Меняем «Allow Full Search» на True, чтобы разрешить поиск по фрагменту (например, марки машины).



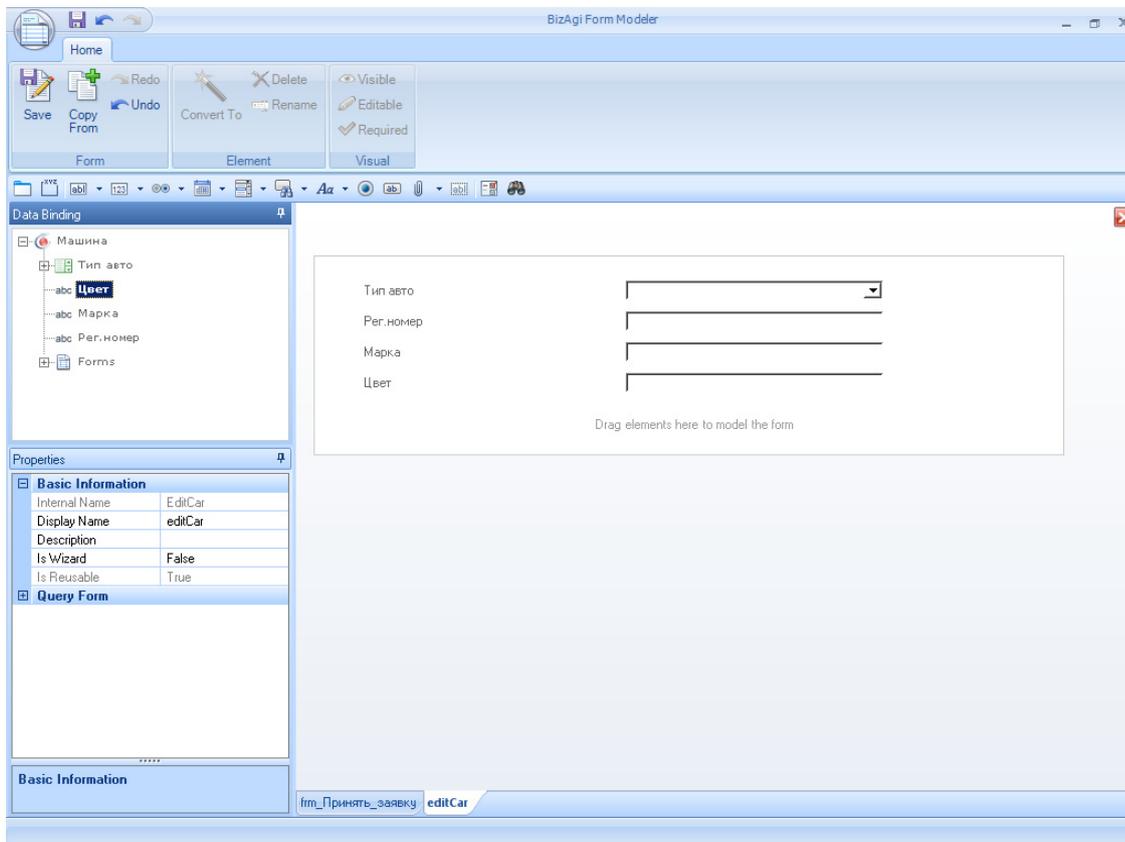
Разрешаем добавление новых записей в справочник:



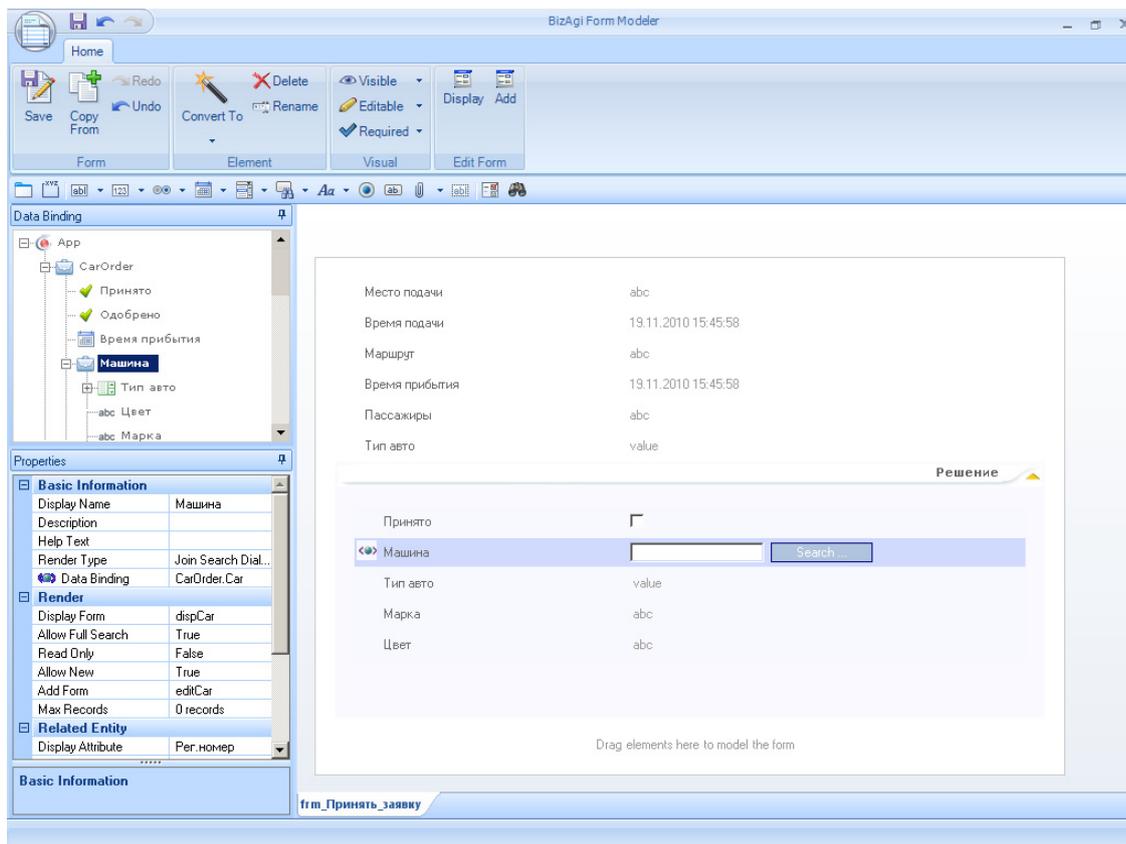
И создаем вторую вспомогательную форму, через которую будем вводить новые автомобили:



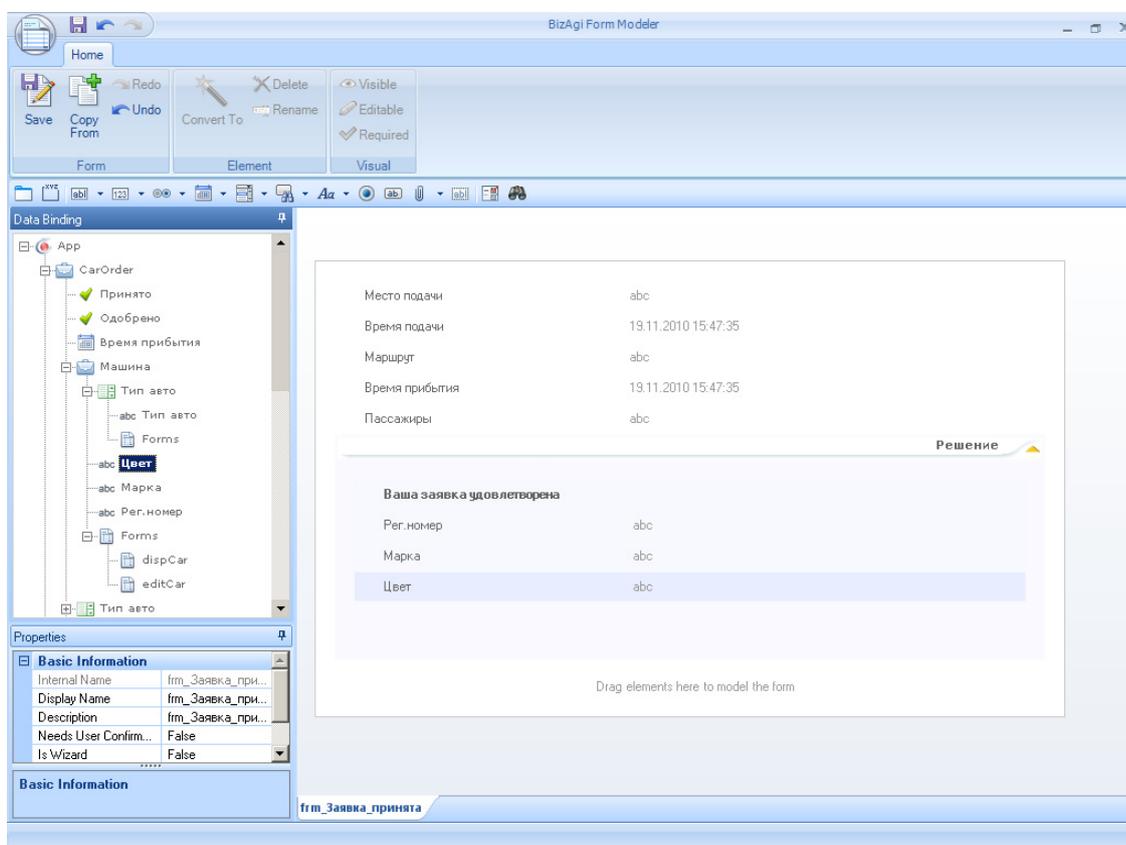
Присваиваем форме имя «editCar» и набрасываем атрибуты. Для выхода щелкаем по красному крестику в правом верхнем углу рабочего поля формы:



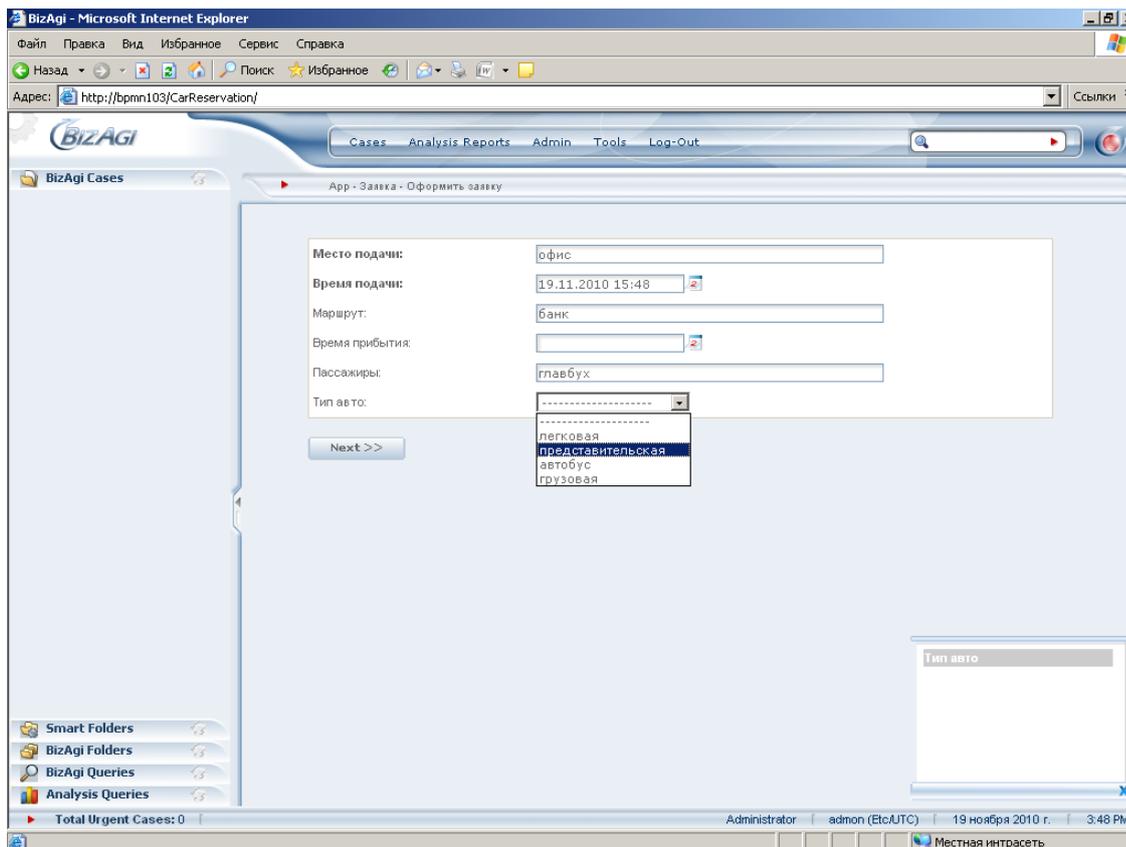
Форма задачи «Принять заявку» готова, выходим с сохранением.



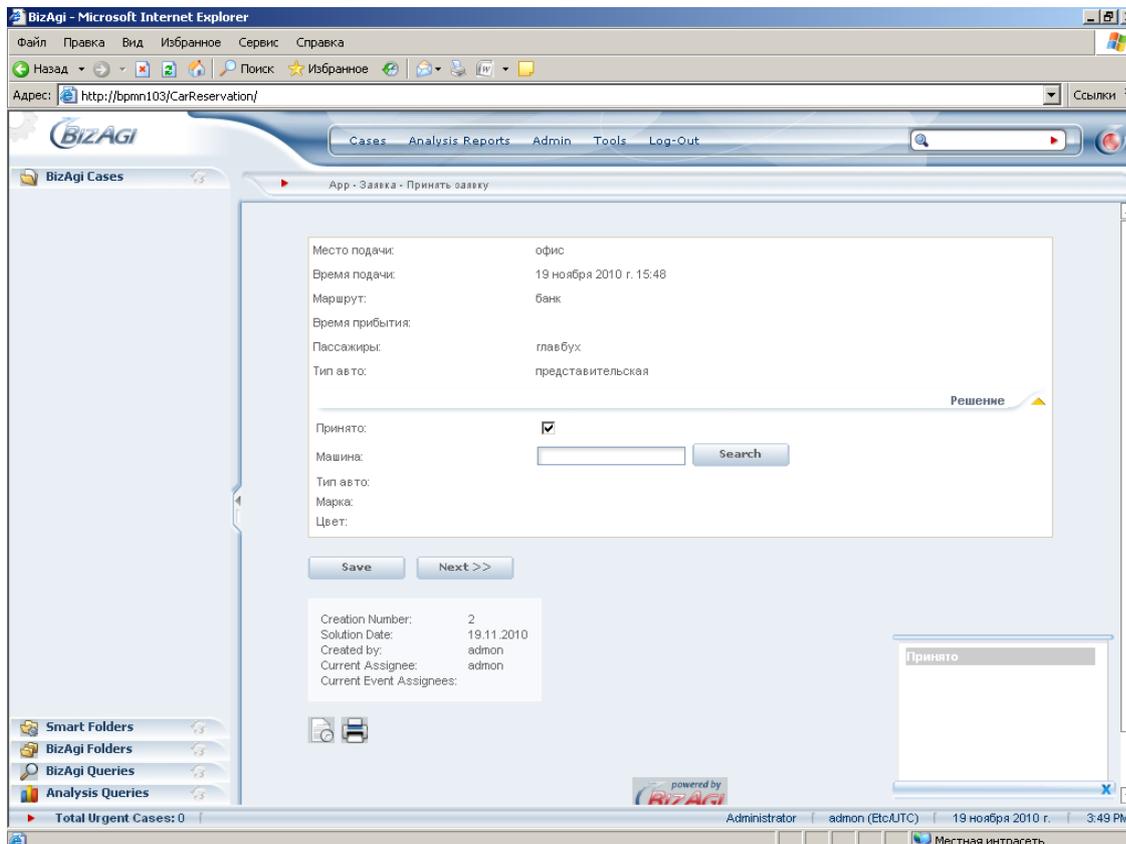
Форма задачи «Заявка принята» проще: на ней ничего нельзя вводить, а надо только сообщить заказчику данные о назначенном автомобиле. Набрасываем на форму атрибуты справочника автомобилей и запрещаем их изменение:



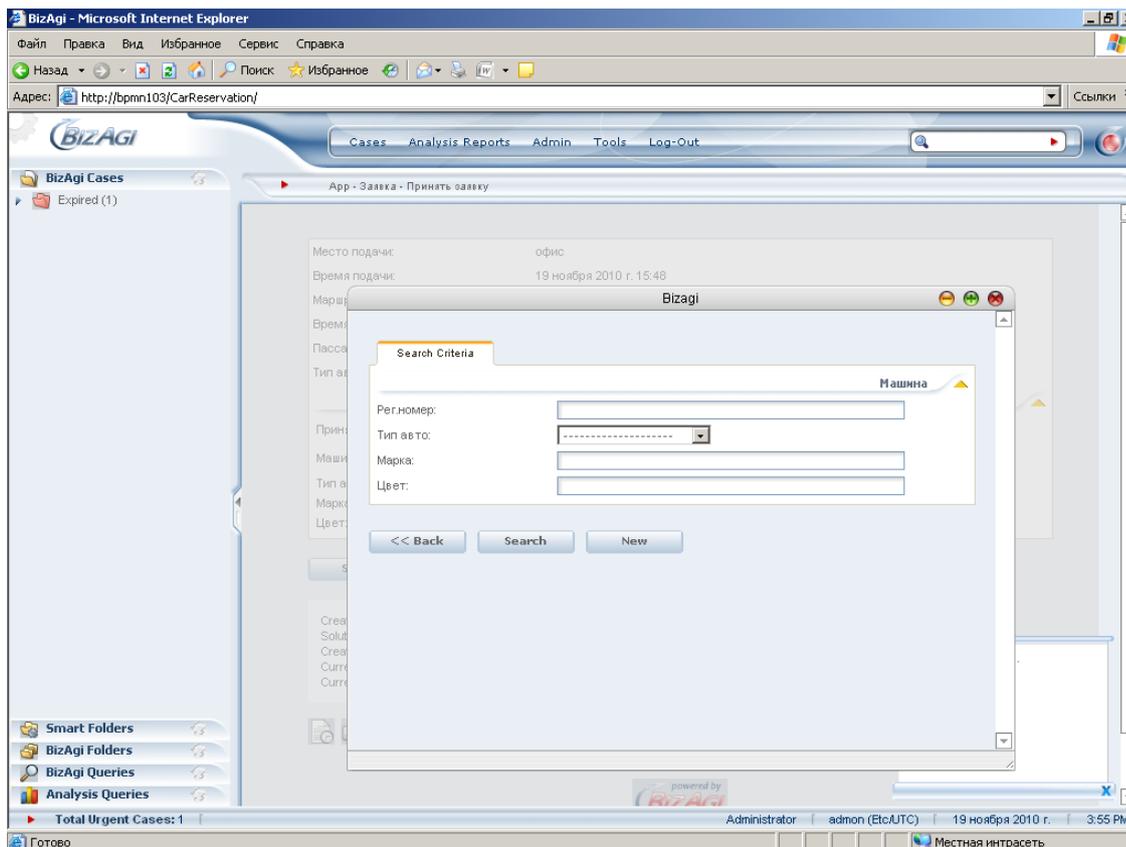
Возвращаемся в портал и запускаем новый экземпляр процесса. На первом шаге выбираем желаемый тип автомобиля:



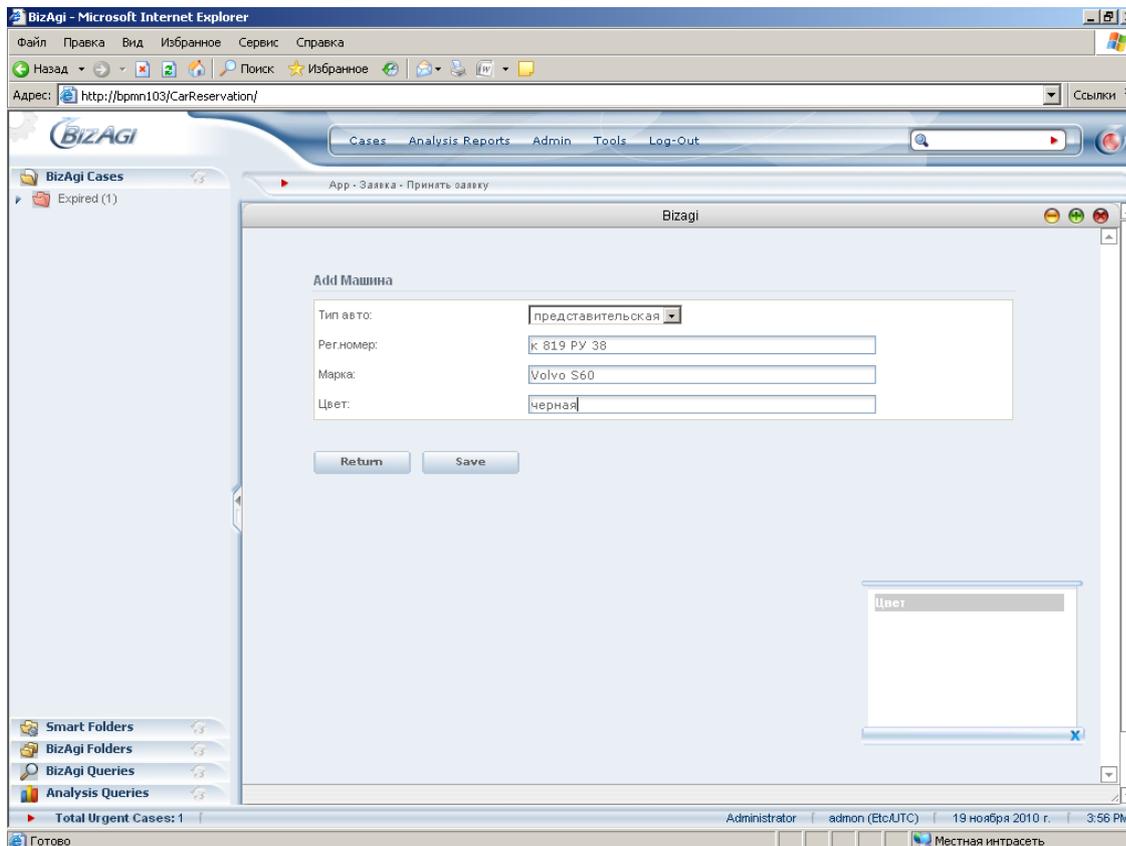
Одобряем заявку и переходим к задаче «Принять заявку». Щелкаем «Search» для выбора автомобиля:



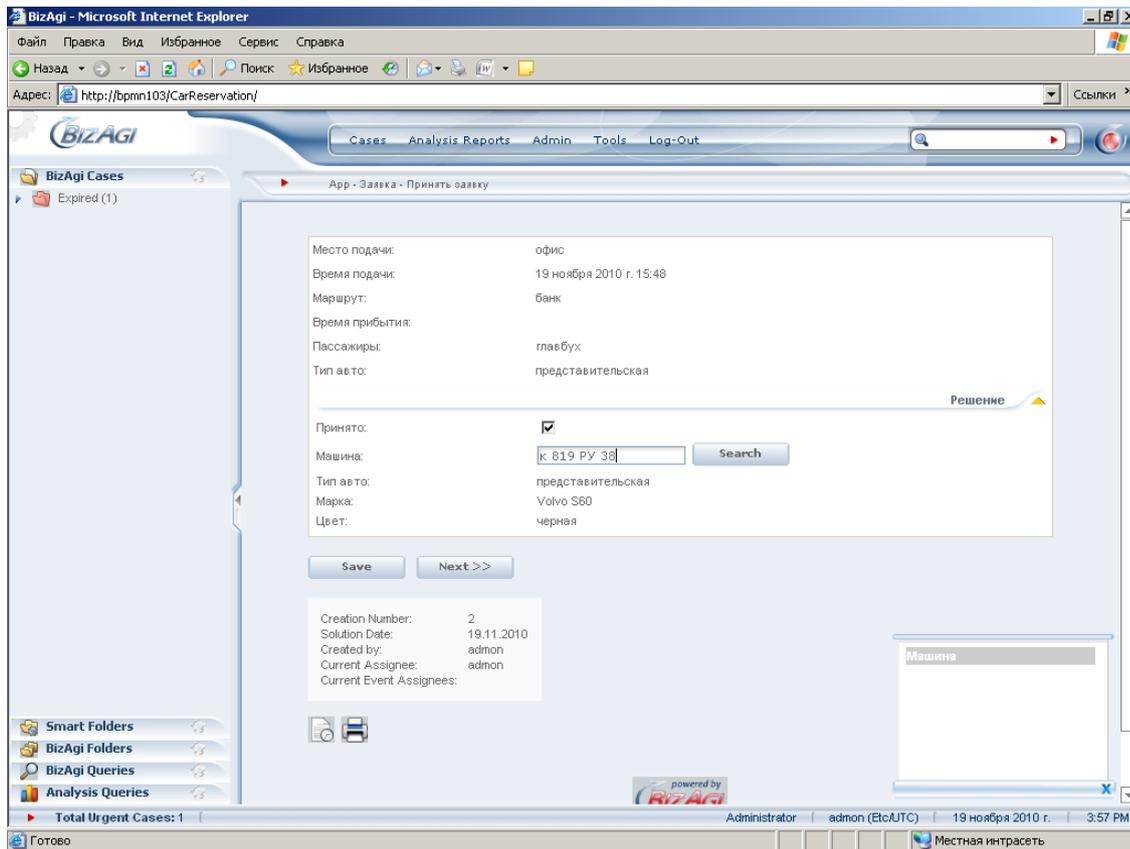
На экране появляется форма поиска. Поскольку ни одного автомобиля в справочнике пока нет, щелкаем «New»:



Вводим данные автомобиля и сохраняем – «Save».

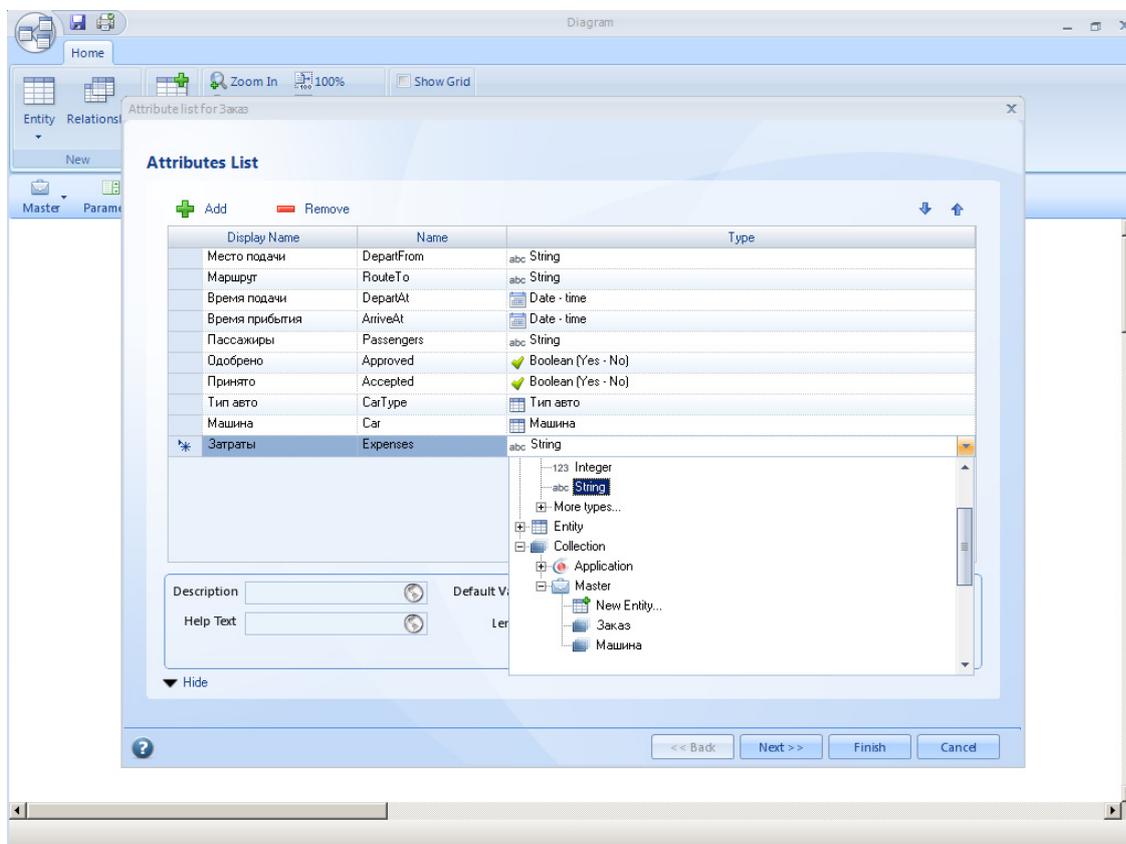


Таким образом мы назначили автомобиль для выполнения данной заявки.

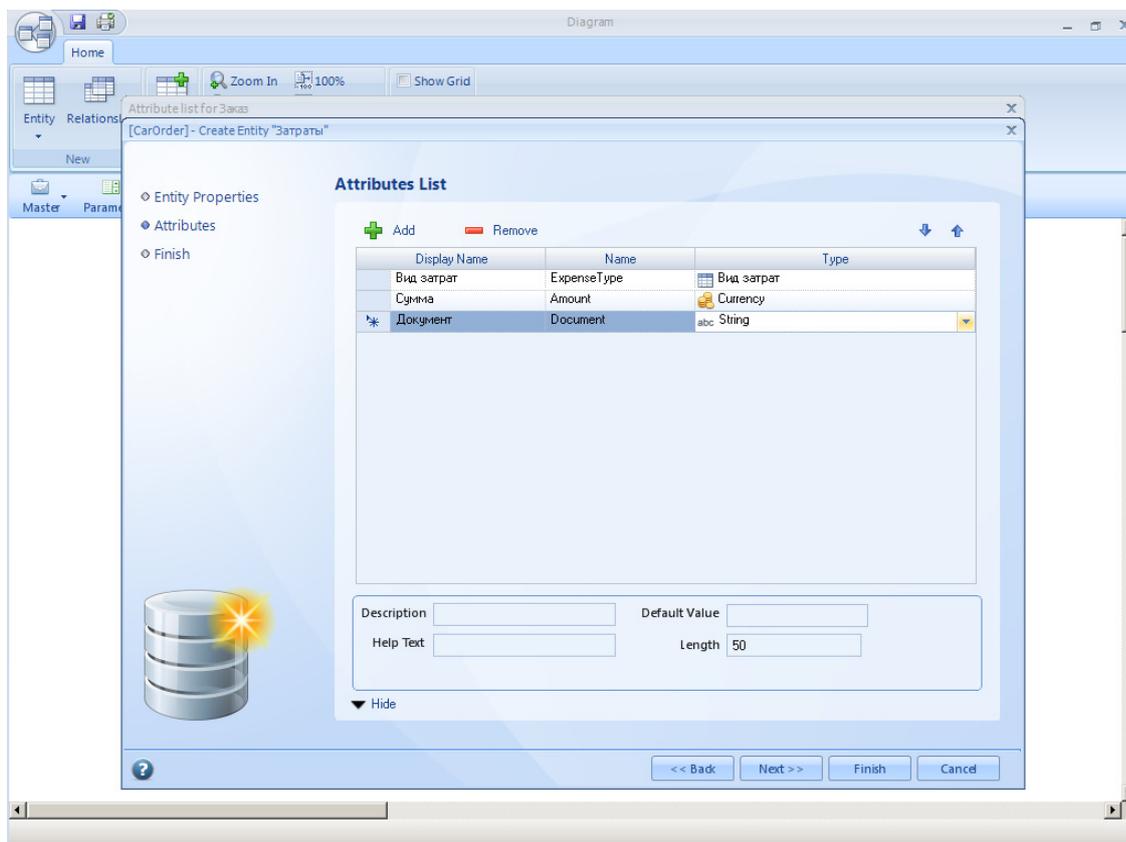


10. Коллекции

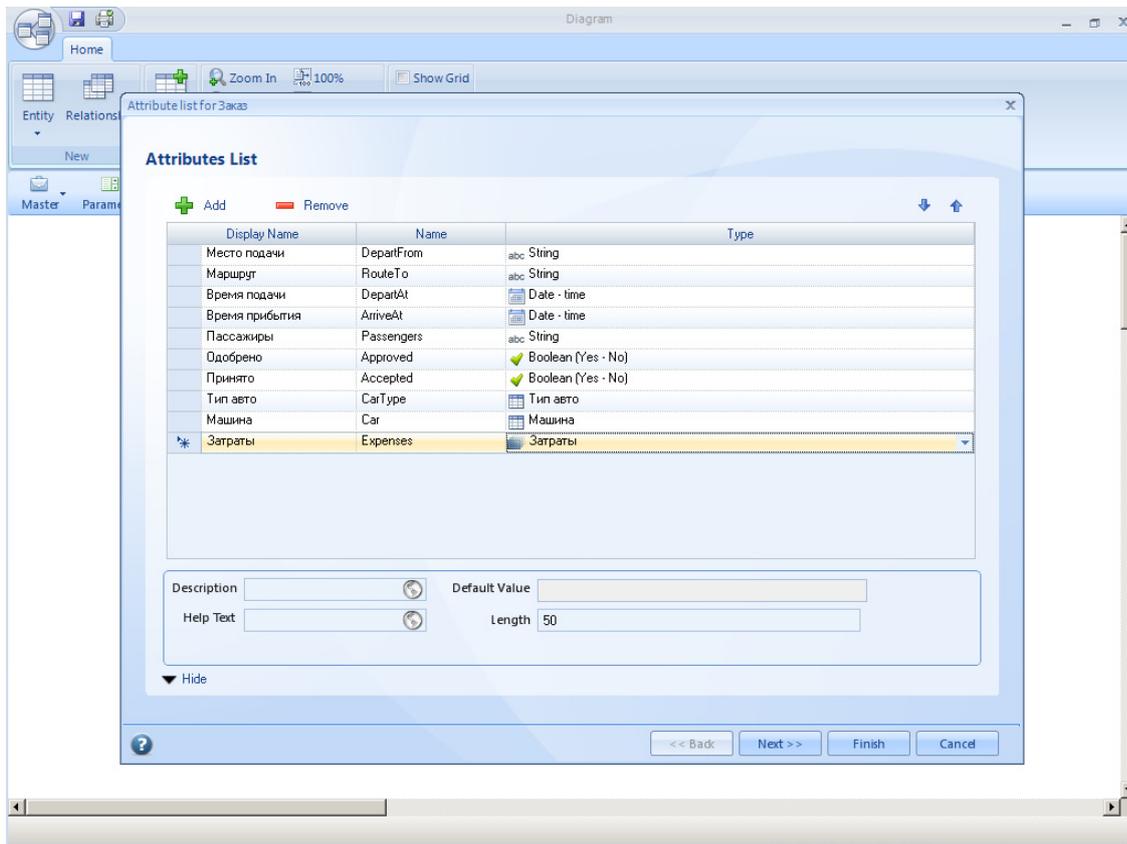
Следующий распространенный сценарий – переход по иерархии не вверх, как в случае справочника, а вниз по иерархии. Например, в рассматриваемом процессе мы можем захотеть ввести прямые затраты на выполнение рейса: топливо, парковка и т.п. Для этого нам понадобятся две таблицы: справочник видов затрат и фактических затрат. Создаем атрибут «Затраты» и выбираем для него тип «Collection» - «Master» - «New Entity...»:



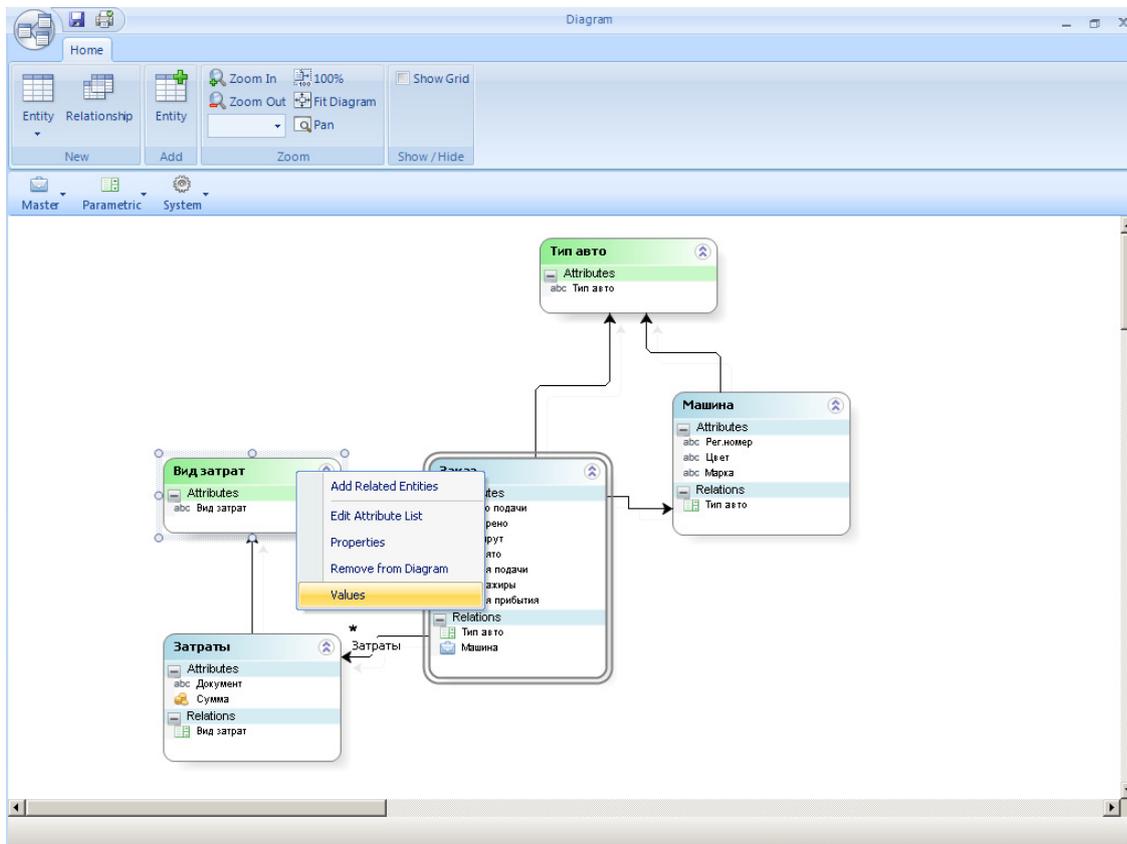
Вводим атрибуты, попутно создаем справочник «Вид затрат» по аналогии со справочником «Тип авто»:



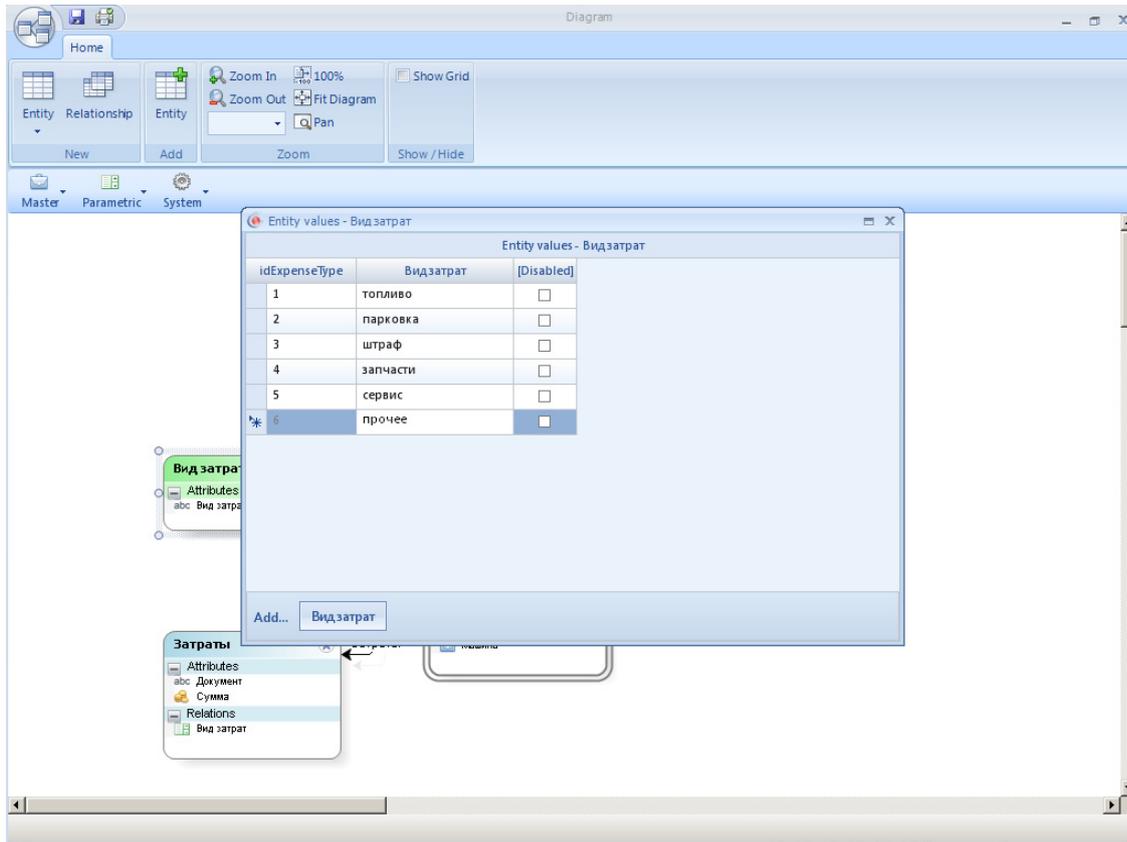
Щелкнув «Finish», сохраняем описание справочника, коллекции и основной таблицы процесса «Заказ».



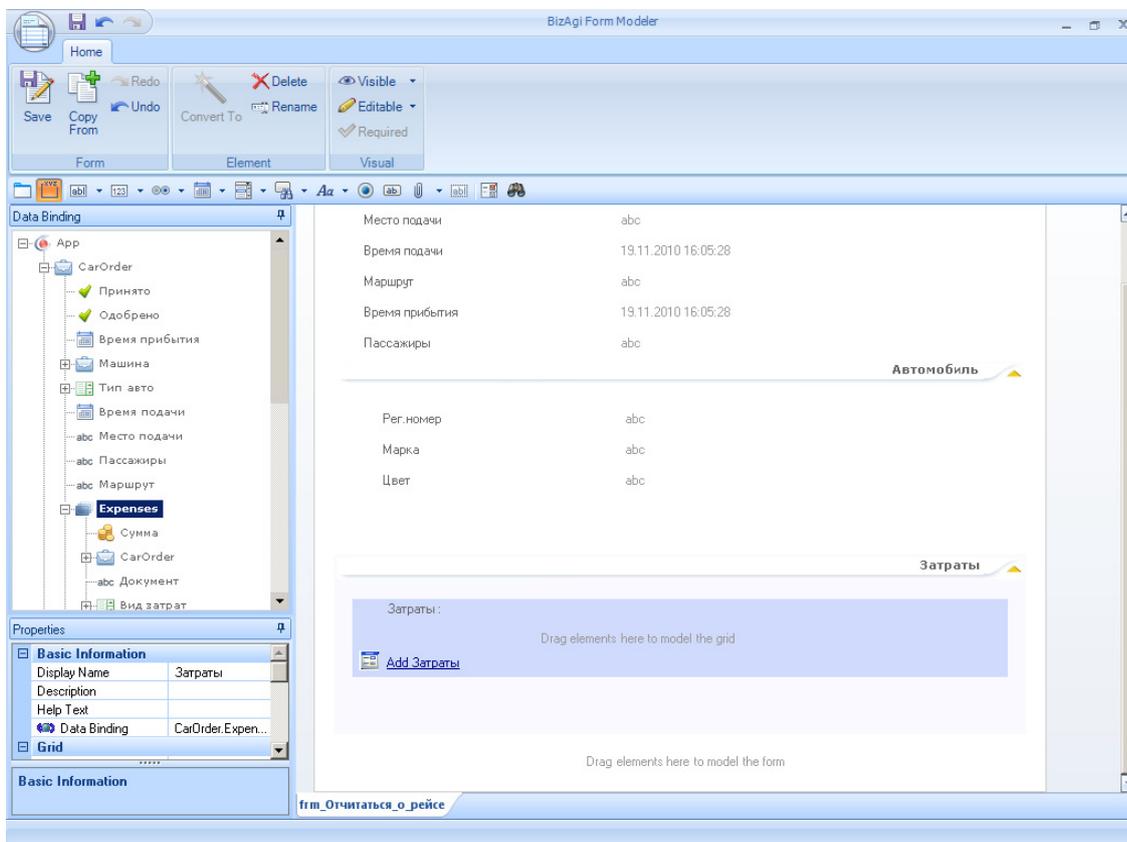
Созданные нами новые таблицы появляются на диаграмме. Справочники типа Parameter окрашены в зеленый цвет, основная таблица процесса выделена двойной рамкой. Заполняем справочник видов затрат:



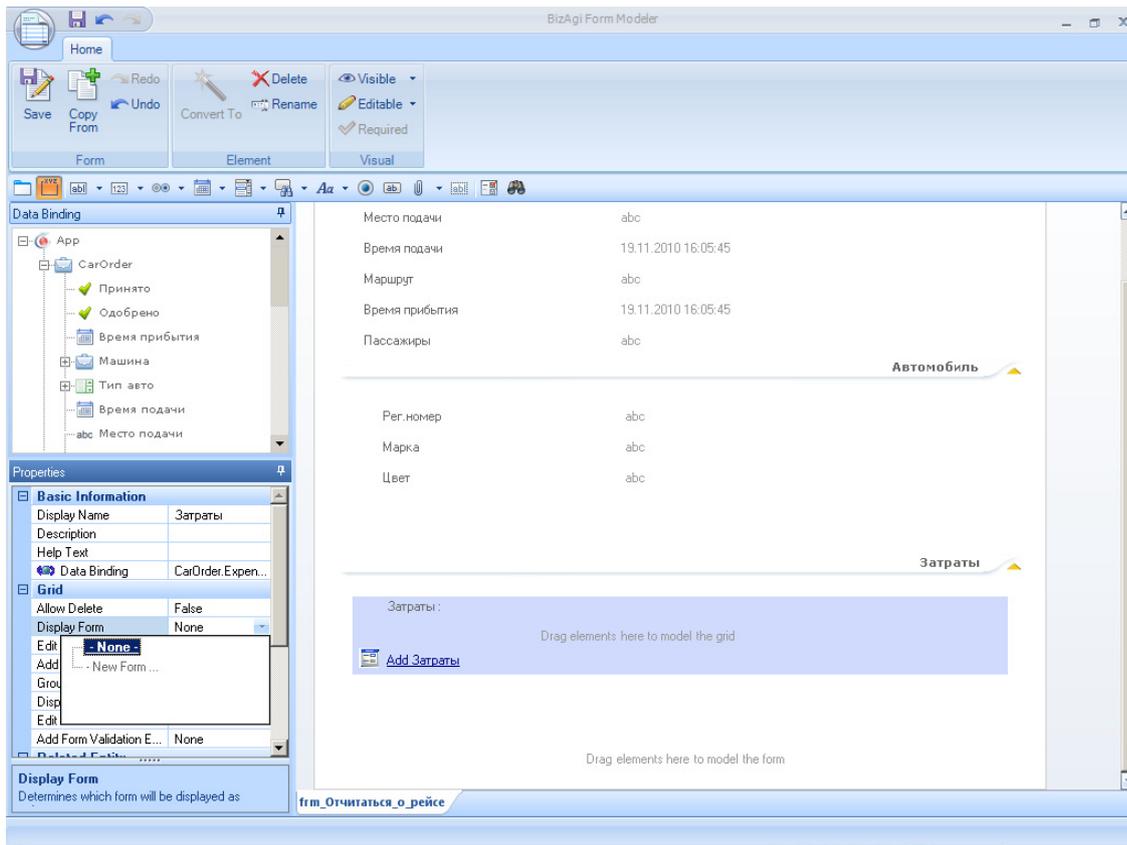
Для этого несколько раз щелкаем кнопку «Вид затрат» и вводим значение:



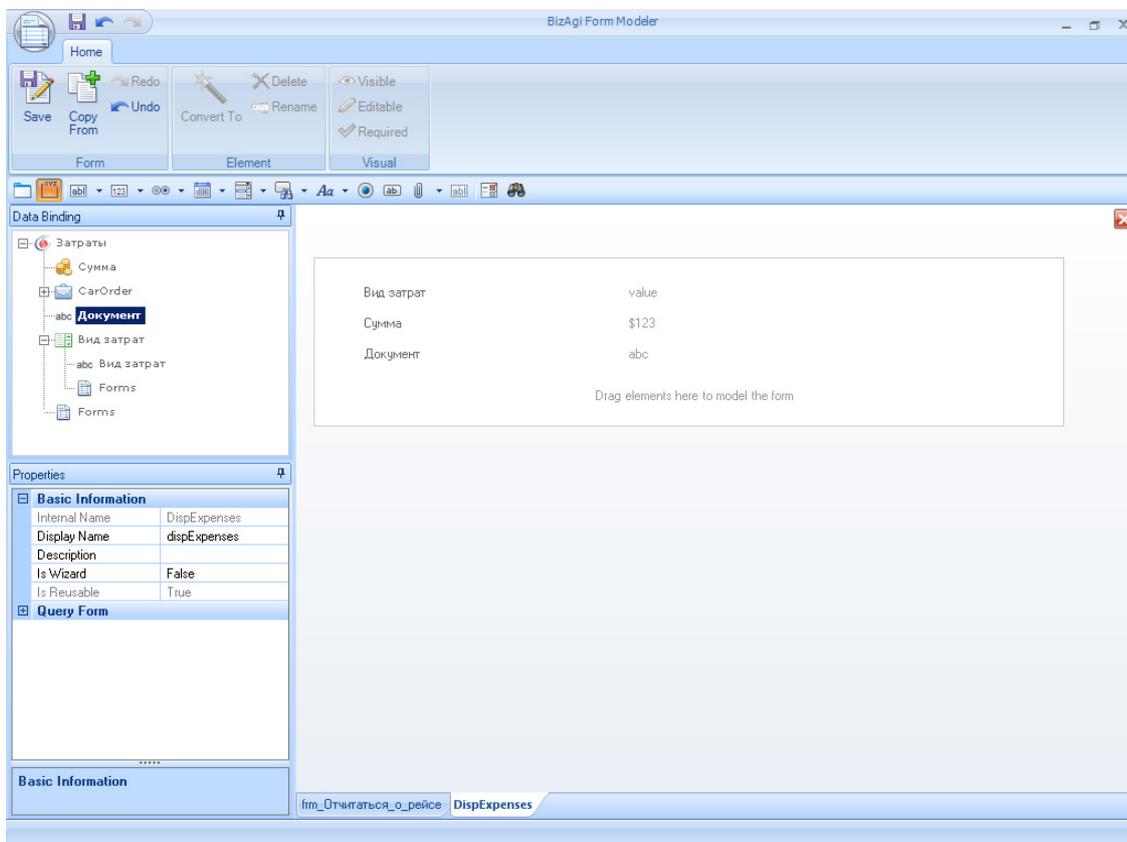
Фактические затраты будут вводиться на форме задачи «Отчитаться о рейсе». В качестве начального приближения для нее берем форму задачи «Заявка принята». Добавляем новый раздел и переносим в него коллекцию, не раскрывая ее:



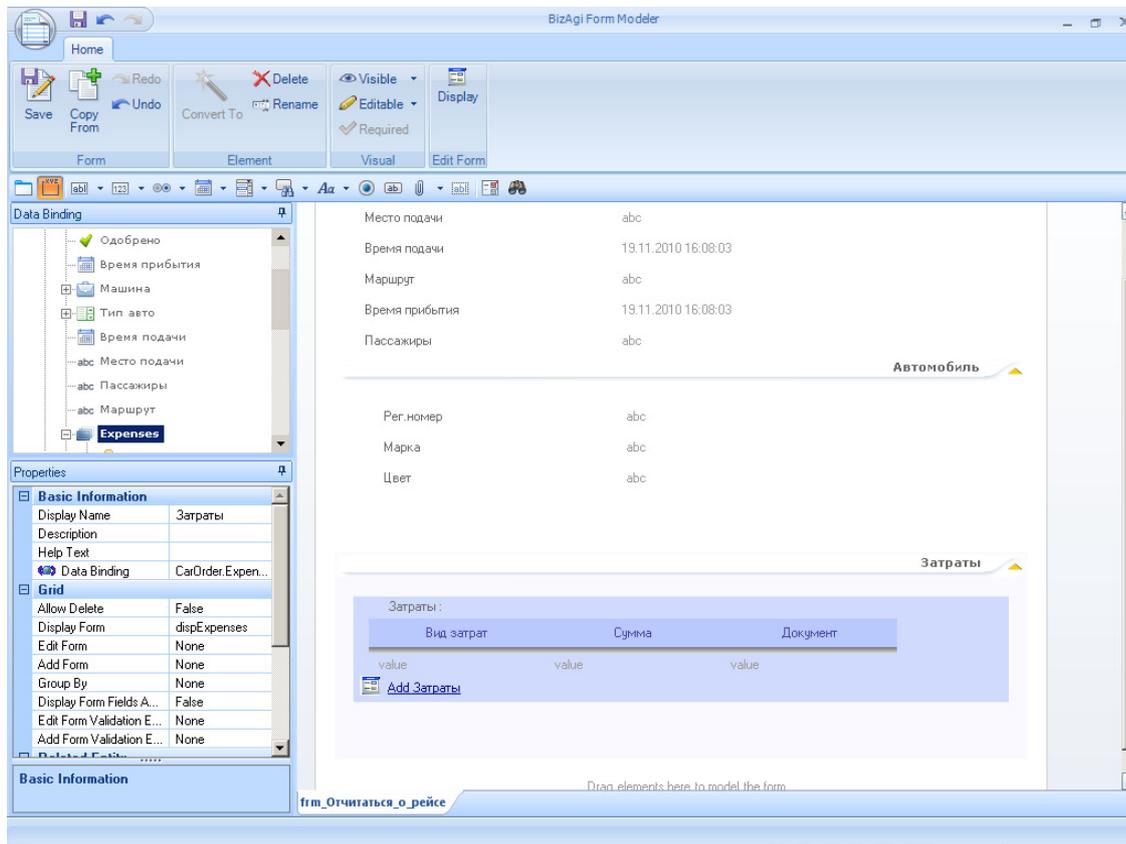
Теперь нам надо создать две вспомогательные формы: для отображения фактических затрат и для ввода. Для отображения выбираем «New Form» из выпадающего списка для «Display Form»:



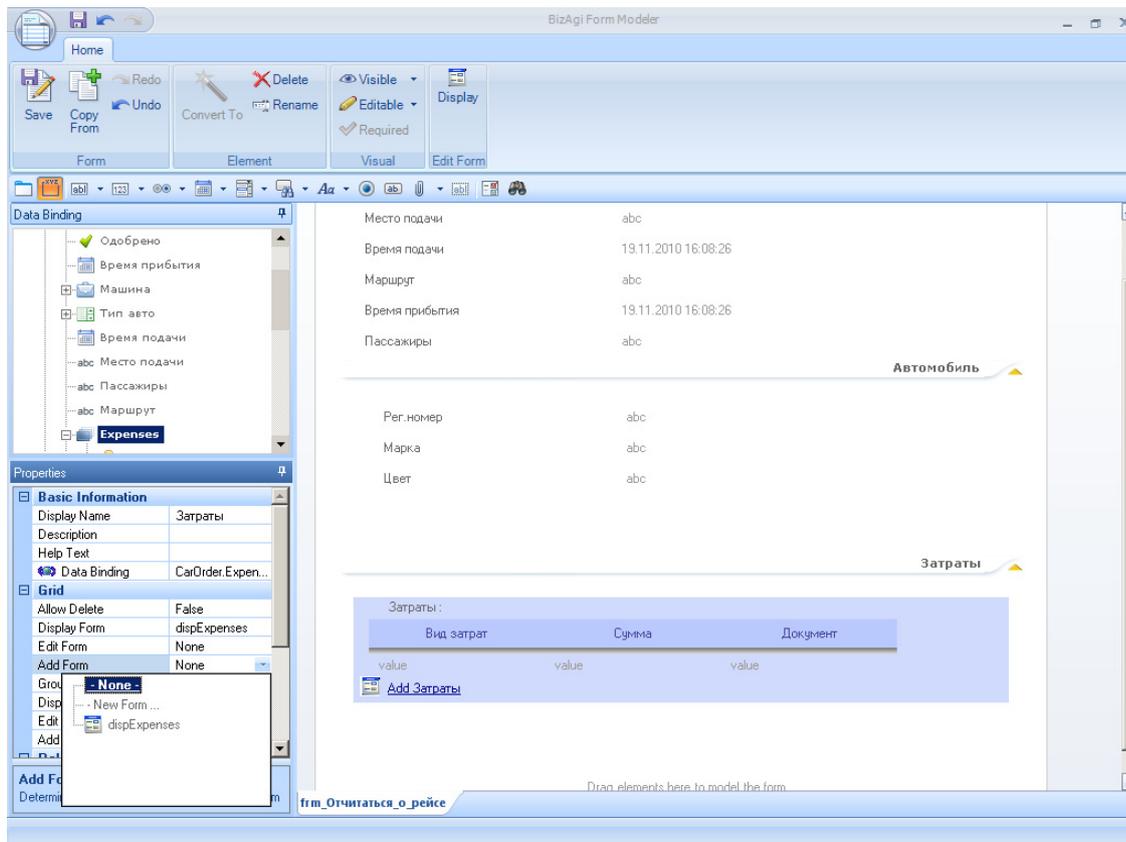
Даем форме имя «dispExpenses», набрасываем атрибуты и делаем их недоступными для редактирования. Для выхода щелкаем по красному крестику в правом верхнем углу рабочего поля формы:



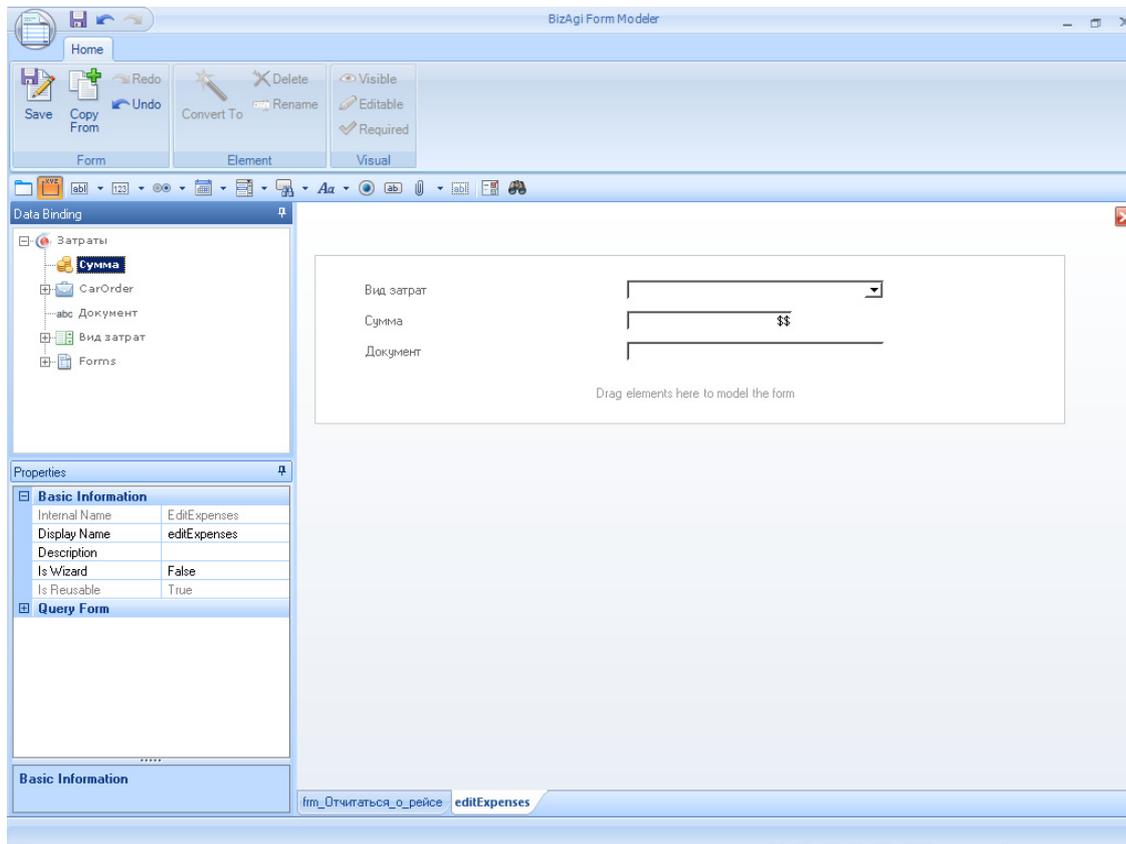
На основной форме появилась таблица затрат:



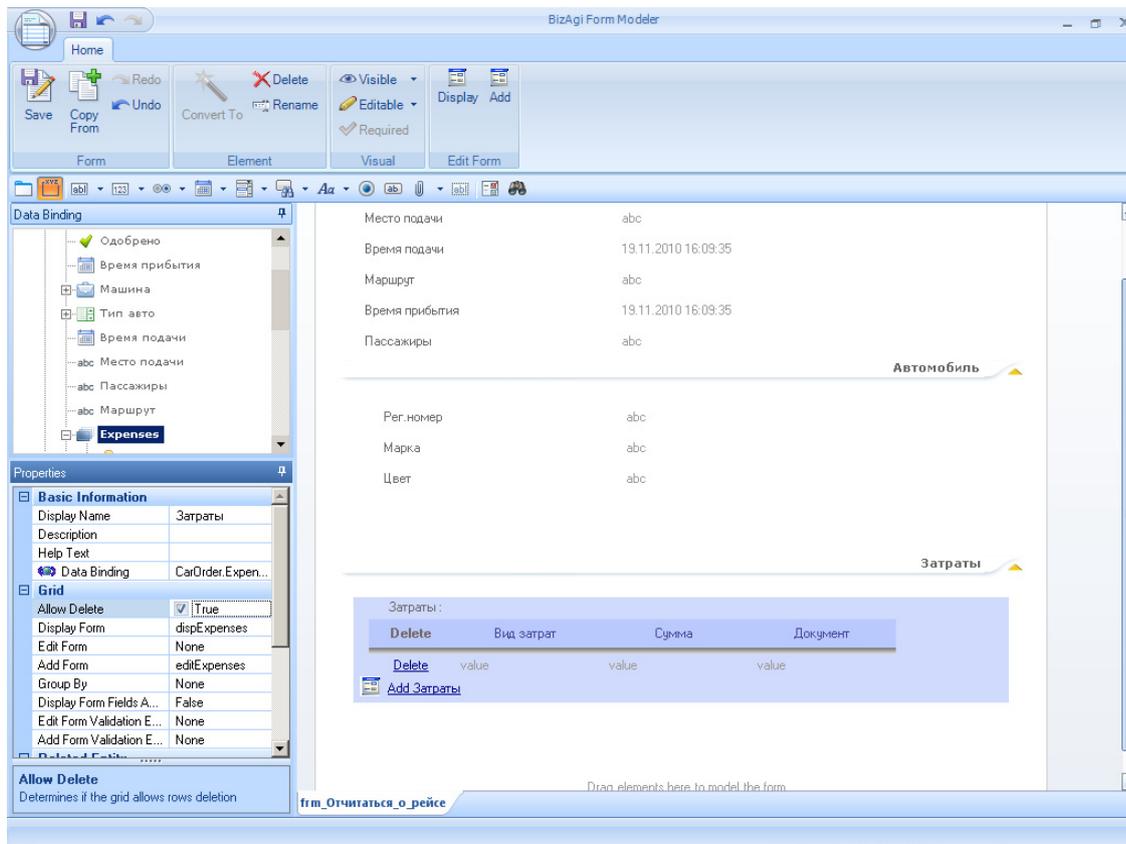
Теперь создаем вторую вспомогательную форму для ввода затрат:



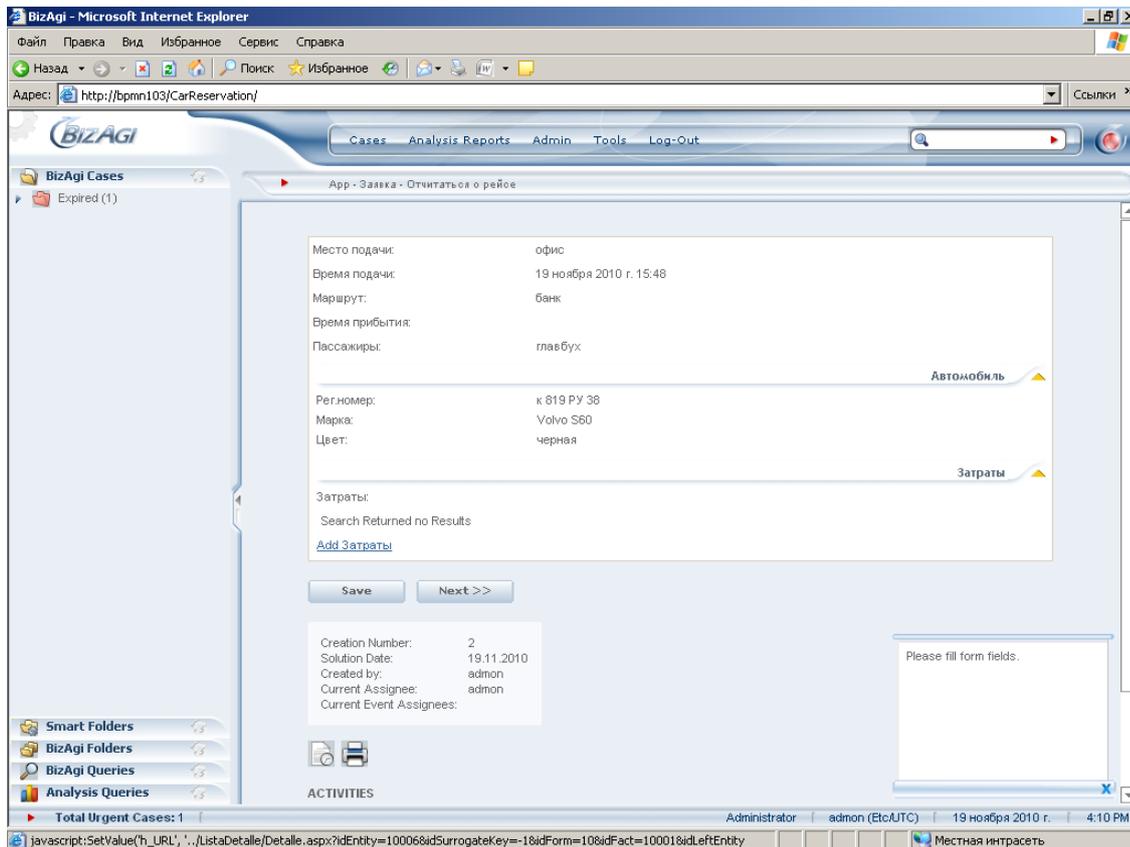
Все аналогично, за исключением того, что поля открыты для редактирования:



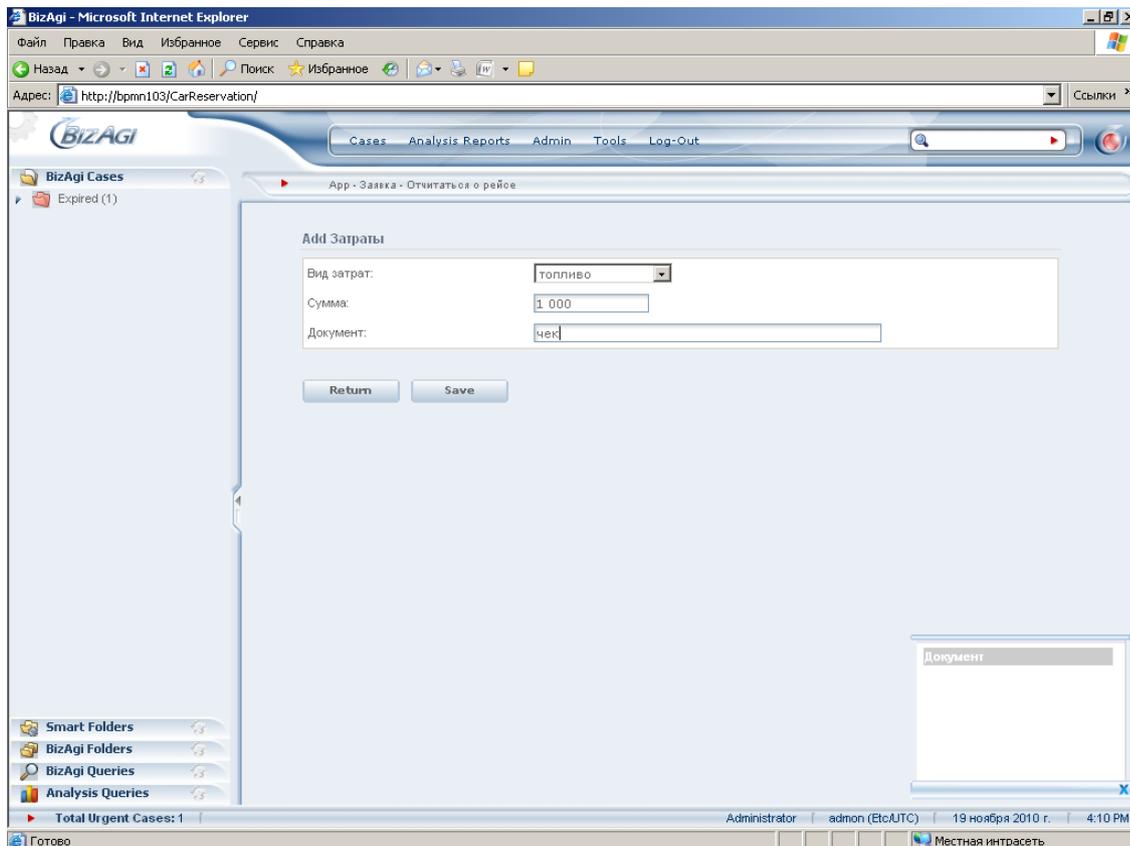
Щелкнув по «Allow Delete», разрешаем не только добавлять, но и удалять затраты:



Посмотрим как форма выглядит на деле. Для ввода затрат щелкаем «Add Затраты»:



Вводим требуемую информацию:



При желании регистрируем другие затраты:

The screenshot shows the BizAgI web application interface in Microsoft Internet Explorer. The browser address bar shows <http://bpnm103/CarReservation/>. The application header includes navigation links: Cases, Analysis Reports, Admin, Tools, Log-Out. The main content area displays a reservation form with the following details:

- Место подачи: офис
- Время подачи: 19 ноября 2010 г. 15:48
- Маршрут: банк
- Время прибытия:
- Пассажиры: главбух

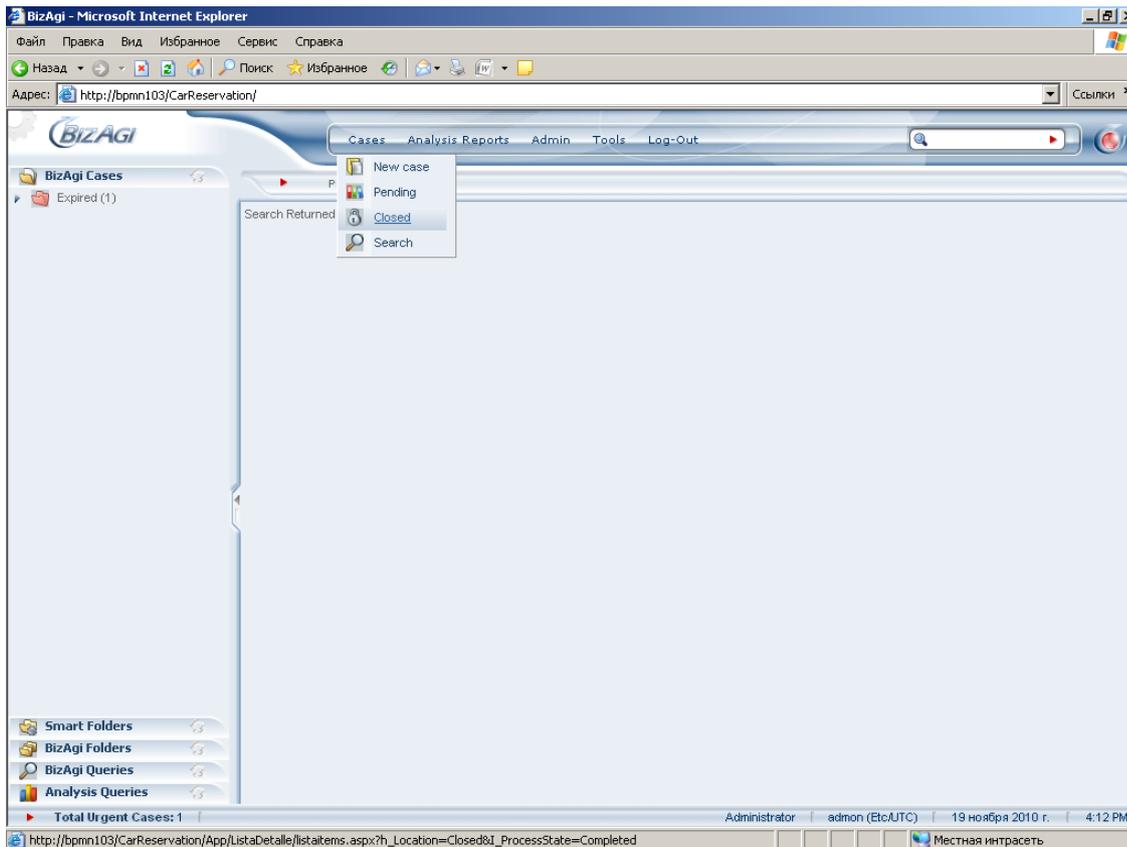
Below these details, there are expandable sections for 'Автомобиль' and 'Затраты'. The 'Затраты' section contains a table:

Delete	Вид затрат	Сумма	Документ
Delete	топливо	1 000,00р.	чек
Delete	парковка	100,00р.	чек

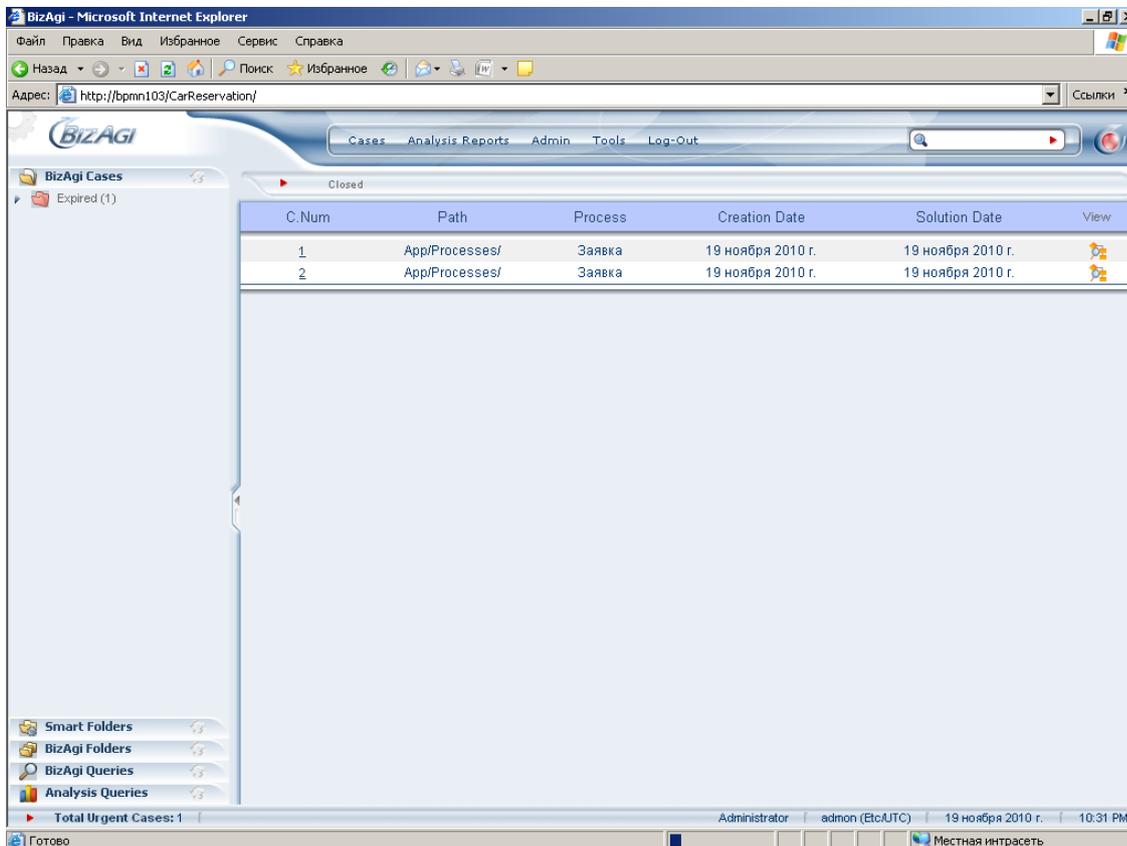
Below the table is a link 'Add Затраты'. At the bottom of the form are 'Save' and 'Next >>' buttons. A small error message box on the right says 'Please fill form fields.' In the bottom left, there is a 'Smart Folders' section with links for 'BizAgI Folders', 'BizAgI Queries', and 'Analysis Queries'. The bottom status bar shows 'Total Urgent Cases: 1', user 'Administrator', session 'admon (EtcAUTC)', date '19 ноября 2010 г.', time '4:11 PM', and 'Местная интрасеть'.

11. Мониторинг

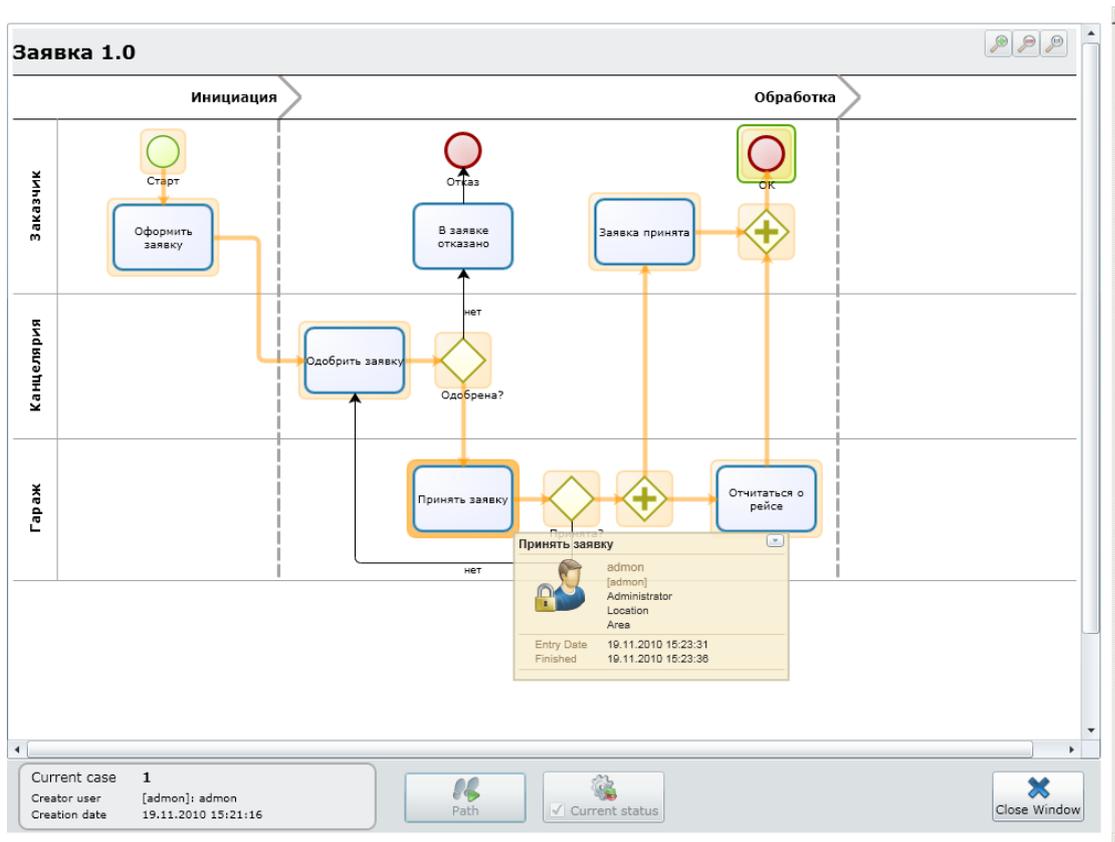
Через портал можно анализировать состояние как исполняющихся, так и завершенных процессов. Например, для просмотра истории выполнения завершенных процессов щелкаем сначала «Cases» - «Closed»:



а затем на иконку в правом столбце списка:



По щелчку «Path» система демонстрирует последовательность выполнения, а если курсор навести на задачу, выводит информацию о том, кто и когда ее выполнил:

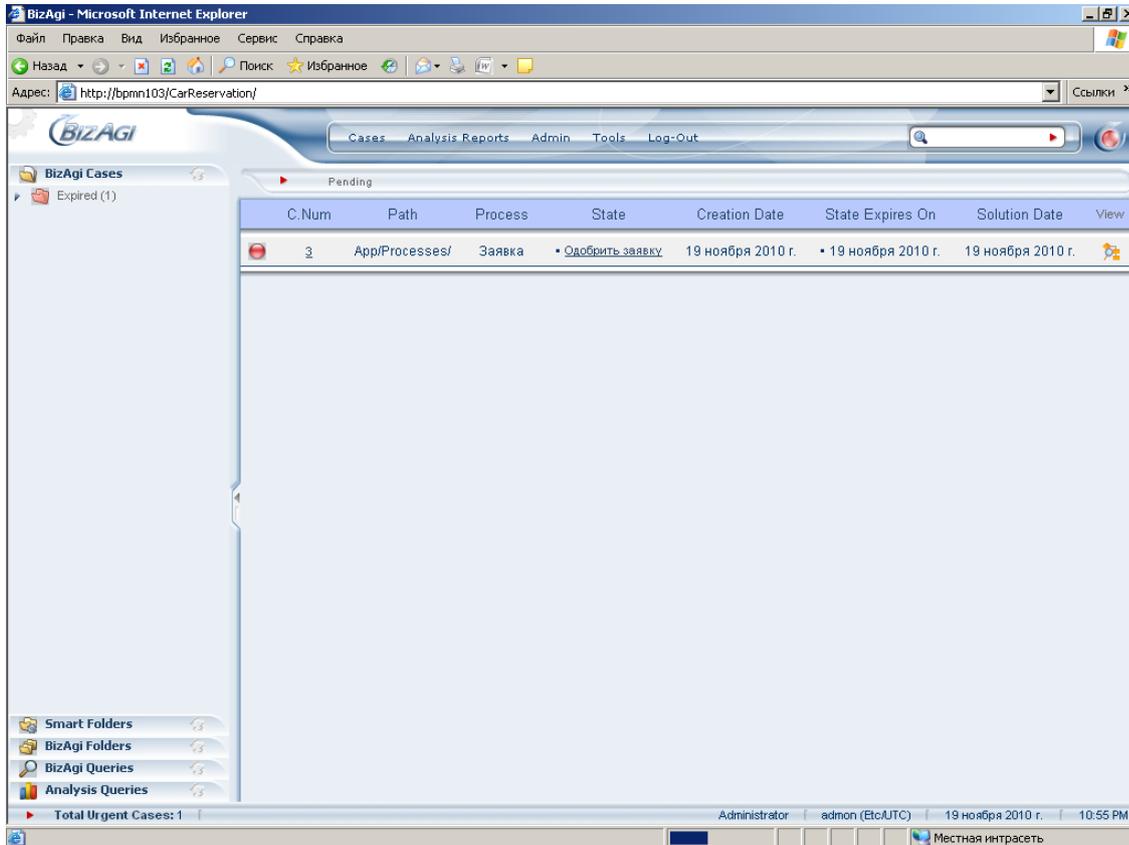


Для отображения этой информации система предложит установить программное обеспечение Microsoft Silverlight, если оно не установлено.

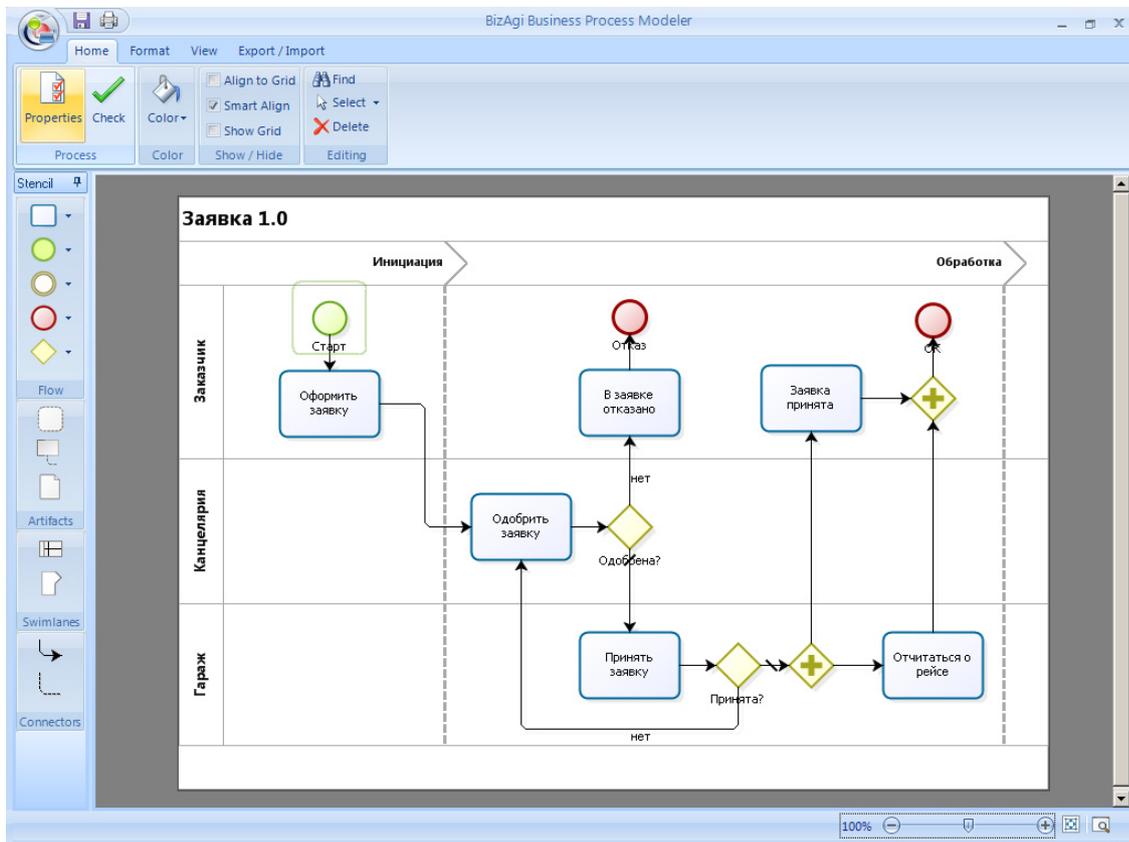


12. Сроки

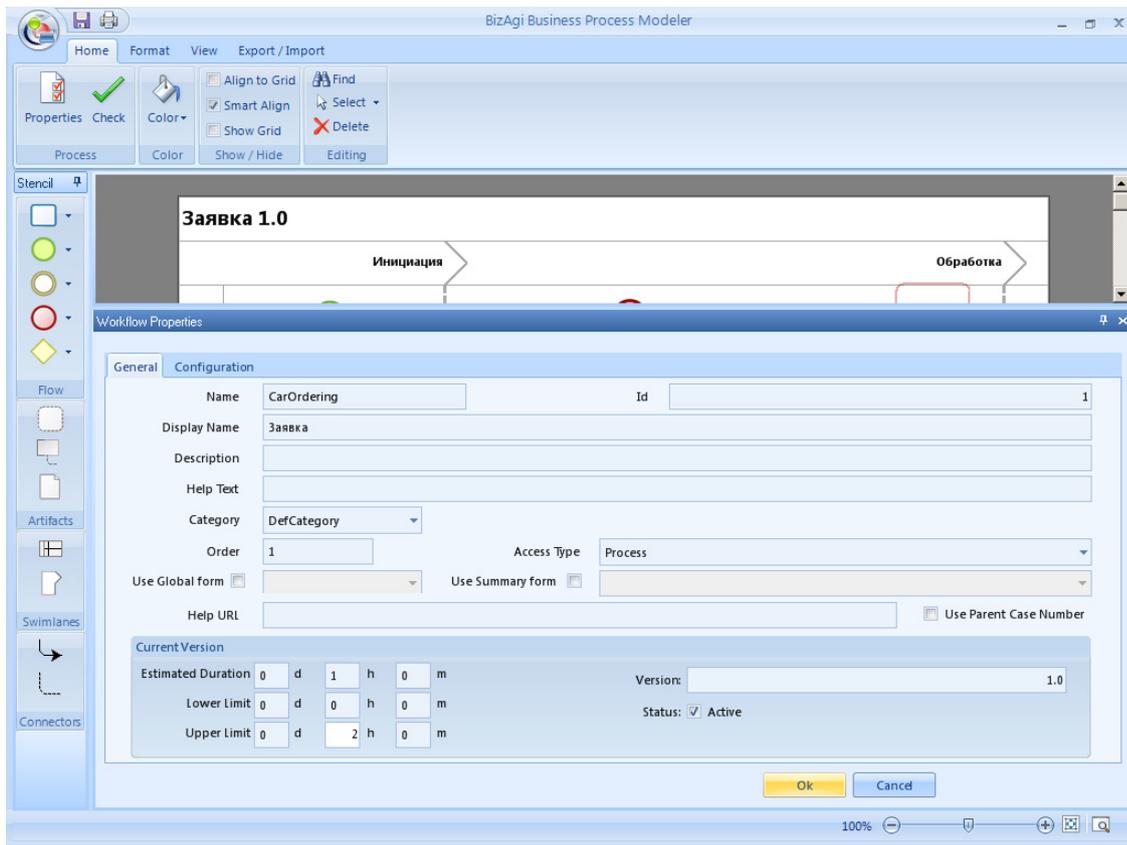
Обратите внимание на красный индикатор в списке задач. Он означает, что задача просрочена. Желтый означает, что срок истекает сегодня, зеленый – что завтра или позже. До сих пор все задачи, которые мы видели, были помечены как просроченные. Получилось так из-за того, что мы не установили нормативный срок исполнения, а по умолчанию он равен нулю. Вот и получилось, что все задачи стали просроченными.



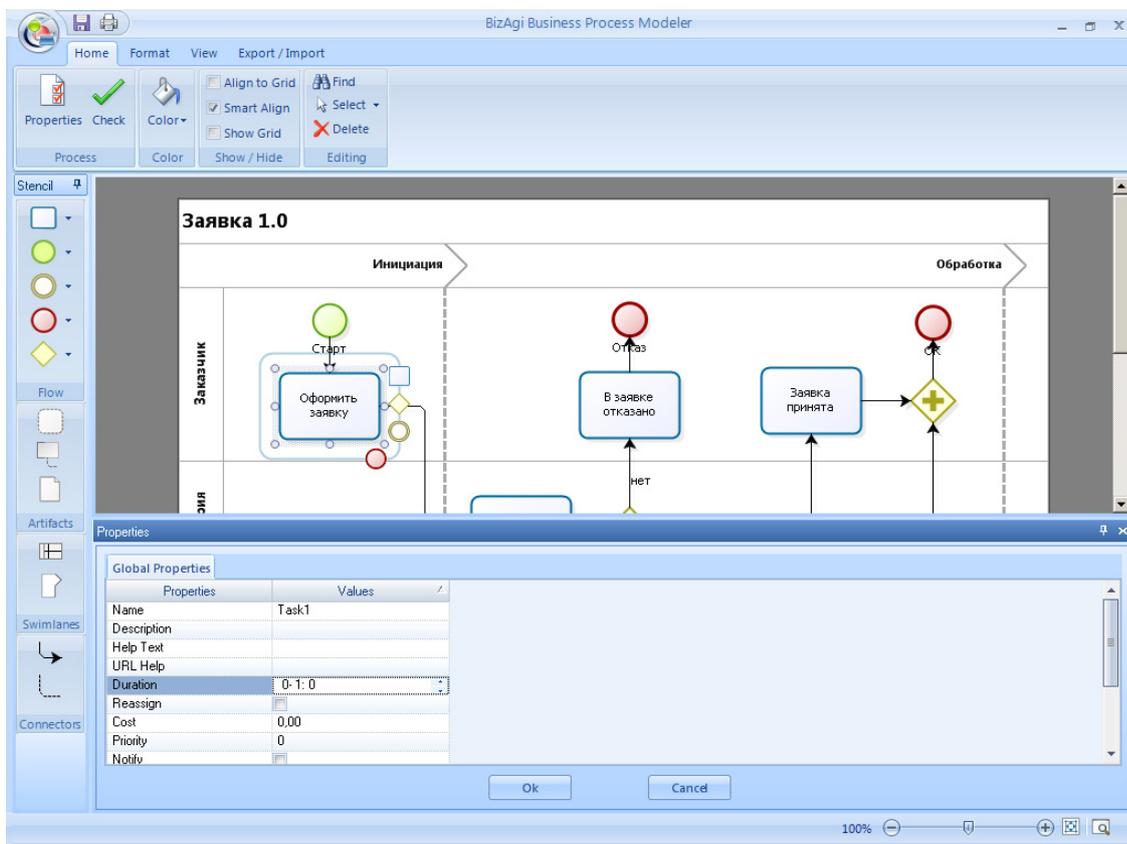
Чтобы это исправить, вернемся к схеме процесса. Сначала щелкнем по «Properties»:



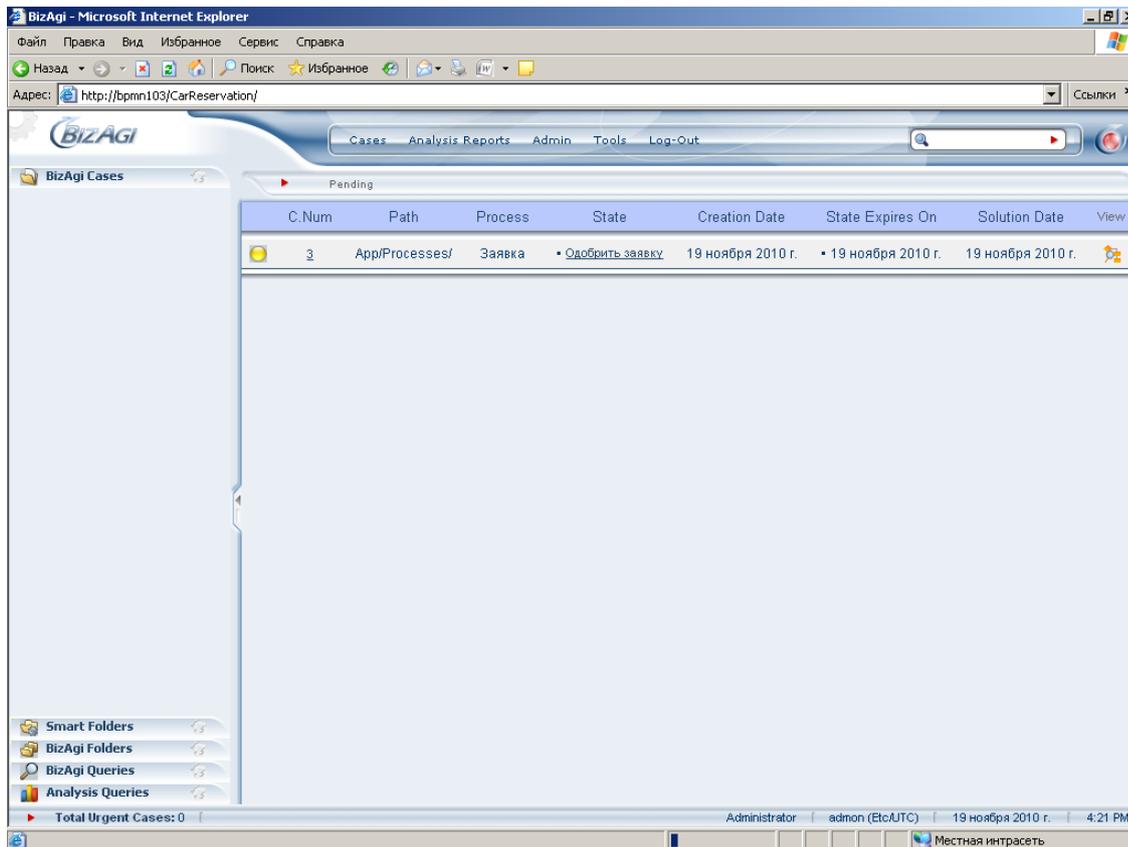
Здесь можно задать ожидаемую («Estimated Duration»), минимальную («Lower Limit») и максимально допустимую («Upper Limit») продолжительности:



Чтобы задать продолжительность задачи, щелкните по ней правой кнопкой и выберите «Properties». Первое число – дни, второе – часы, третье – минуты. Задайте для всех задач продолжительность 1 час:



Теперь только что созданные задачи будут помечаться желтым, а через час они будут становиться красными.



Поздравляем, вы успешно справились с лабораторной работой по курсу «ВРМН103 – Исполняемый ВРМН», часть 1 «Оркестровка».