

Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра системного программирования

# РЕАЛИЗАЦИЯ ГЕНЕРАЦИИ ИСХОДНОГО КОДА БИЗНЕС-ПРОЦЕССОВ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ДИАГРАММ BPMN В ТЕХНОЛОГИИ REAL-IT/.NET

Дипломная работа студента 545 группы

Мокаева Руслана Назировича

Научный руководитель	..... / подпись /	д.ф.-м.н., проф. Терехов А.Н.
Рецензент	..... / подпись /	Нестеров А.В.
“Допустить к защите” заведующий кафедрой,	..... / подпись /	д.ф.-м.н., проф. Терехов А.Н.

Санкт-Петербург

2013

Saint-Petersburg State University  
Mathematics and Mechanics Faculty

Software Engineering Department

**Implementation of informational system  
business processes source code generation  
based on BPMN diagrams in REAL-IT/.NET**

Graduate paper by

Ruslan Mokaev

Scientific advisor ..... Professor A.N. Terekhov

Reviewer ..... A.V. Nesterov

“Approved by” ..... Professor A.N. Terekhov  
Head of Department

Saint-Petersburg

2013

## Оглавление

<b>Введение</b> .....	4
<b>Постановка задачи</b> .....	6
Выбор редактора диаграмм.....	6
Реализация необходимой логики в библиотеках поддержки исполнения.....	6
Внесение изменений в генераторы и шаблоны экранных форм и базы данных.....	6
Апробация полученных генераторов на примере ИС.....	7
<b>Обзор редакторов диаграмм</b> .....	8
Bizagi Process Modeler.....	8
Signavio Process Editor .....	8
Yaoqiang BPMN Editor.....	8
Microsoft Visio .....	8
Дизайнер ELMA .....	9
ARIS Express.....	9
Modelio.....	9
Enterprise Architect .....	9
QReal.....	10
<b>Обзор языков моделирования бизнес-процессов</b> .....	11
BPEL.....	11
BPMN .....	12
<b>Описание решения</b> .....	14
Выбор редактора диаграмм.....	14
Генерация кода бизнес-процессов.....	16
Создание служебных состояний.....	16
Работа со статическими классами, сохраняющими состояние уникальных объектов.....	17
Получаемые из диаграммы бизнес-процессов ограничения и действия .....	18
Изменение библиотек поддержки исполнения .....	19
<b>Апробация</b> .....	20
<b>Заключение</b> .....	25
<b>Направления дальнейшей работы</b> .....	25
<b>Список литературы</b> .....	26

## **Введение**

Такой сегмент рынка разработки программного обеспечения, как создание систем, направленных на сбор, хранение и обработку информации, всегда был одним из самых популярных. Как и все остальное в мире ИТ он развивается стремительно, вследствие чего разрабатываются разнообразные подходы для автоматизации процесса разработки. Одним из них является разработка с использованием объектно-ориентированного визуального моделирования с использованием CASE-средств для создания моделей системы [2].

На кафедре системного программирования Санкт-Петербургского государственного университета была разработана технология REAL-IT, основанная на использовании объектно-ориентированного CASE-пакета REAL, также разработанного на математико-механическом факультете СПбГУ. Технология REAL-IT позволяет создавать и поддерживать приложения, ориентированные на сбор, хранение и обработку данных.

Разработка информационных систем на тот момент осуществлялась на языке Visual Basic 6.0, что вводило ограничения на возможности по разработке графического пользовательского интерфейса. Также к проблемам, возникавшим перед пользователями и разработчиками, относилось отсутствие инструментария для установки автоматических обновлений и устаревшая среда разработки. Поэтому в 2007 году в рамках дипломной работы Антона Нестерова был реализован перенос технологии REAL-IT на платформу Microsoft .NET.

В текущей версии REAL-IT все алгоритмы поведения системы, не вытекающие тривиальным образом из модели данных, должны быть запрограммированы вручную и информация о поведении системы, которая может быть получена из соответствующих диаграммах (например, диаграмм

бизнес-процессов) не используется при генерации. Основная сложность в использовании поведенческих моделей заключается в том, что этот код должен учитывать модель данных и должен быть проинтегрирован с кодом, полученным исходя из моделей данных.

Таким образом, генерация кода на основе нескольких независимых моделей, каждая из которых описывает систему со своей точки зрения, является очень важной нерешенной проблемой [1].

## **Постановка задачи**

Задачей данной работы является реализация генераторов исходного кода бизнес-процессов ИС на основе диаграмм BPMN в рамках технологии REAL-IT/.NET. В неё входят следующие подзадачи:

## **Выбор редактора диаграмм**

Редактор диаграмм должен удовлетворять следующим критериям:

1. Возможность связывания диаграмм классов и диаграмм бизнес-процессов. Другими словами, объект данных, участвующий в бизнес-процессе (например, в обмене информацией между участниками бизнес-процесса), должен быть экземпляром некоторого класса или полем некоторого класса из диаграммы классов. Изменения, вносимые в диаграмму классов (например, переименования), должны вноситься и в диаграмму бизнес-процессов;
2. Возможность экспорта спроектированных диаграмм в xml-подобный формат (то есть возможность извлечения информации в удобном виде).

## **Реализация необходимой логики в библиотеках поддержки исполнения**

Необходимо реализовать определенные общие для всех сгенерированных приложений функции и внести соответствующие изменения в библиотеки поддержки исполнения, тем самым уменьшив объем генерируемого кода и улучшив его читаемость.

## **Внесение изменений в генераторы и шаблоны экранных форм и базы данных**

Необходимо внести изменения в текущий генератор, позволяющие генерировать ограничения нескольких типов, которые могут быть получены из диаграммы бизнес-процессов (например, ограничения на создание, редактирование или удаление объектов данных). Также необходимо

генерировать статические классы для участников бизнес-процесса, являющихся уникальными.

### **Апробация полученных генераторов на примере ИС**

Необходимо спроектировать пример, который будет содержать всевозможные элементы нотации BPMN 2.0 и апробировать измененные генераторы.

## Обзор редакторов диаграмм

Основным требованием при выборе редактора диаграмм является наличие следующего функционала:

1. связывание диаграмм бизнес-процессов (спецификация BPMN 2.0) и диаграмм классов UML;
2. экспортирование спроектированной диаграммы в xml-подобный формат.

В процессе работы были проанализированы следующие редакторы:

### **Bizagi Process Modeler**

Свободно распространяемое компанией Bizagi средство, позволяющее создавать диаграммы BPMN. Имеется возможность экспортировать спроектированную диаграмму в формат XPD (XML Process Definition Language).

### **Signavio Process Editor**

Web-based средство моделирования, ориентированное исключительно на бизнес-процессы и поддерживающее стандарт BPMN 2.0. Имеется возможность экспорта диаграмм в формат bpmn 2.0 xml.

### **Yaoqiang BPMN Editor**

Свободно распространяемое и активно развивающееся средство моделирования, поддерживающее стандарт BPMN 2.0. Имеется возможность экспорта в формат bpmn 2.0 xml.

### **Microsoft Visio**

Векторный графический редактор, редактор диаграмм и блок-схем для Windows, поддерживающий стандарт BPMN 2.0. Отсутствует встроенная возможность экспорта спроектированной диаграммы в xml-формат, однако



существует ряд платных плагинов, добавляющих подобную функциональность.

## **Дизайнер ELMA**

Приложение с простым и понятным интерфейсом компании ELMA, поддерживающее стандарт BPMN 2.0 и направленное исключительно на моделирование бизнес-процессов. Входит в бесплатный пакет ELMA.

## **ARIS Express**

Свободное средство моделирования для управления и анализа бизнес-процессов, разработанное компанией Software AG, поддерживающее, в том числе, стандарт BPMN 2.0. Express версия не предоставляет возможность экспорта диаграмм в xml-формат, однако платная версия продукта ARIS Business Architect включает в себя эту функциональность.

## **Modelio**

Инструмент с открытым кодом, поддерживающий стандарты UML2.x и BPMN2.0. Имеется возможность экспортировать диаграммы UML в формат XMI (XML Metadata Interchange - стандарт OMG для обмена метаданными с помощью XML, часто используемый как формат обмена UML- моделями). Возможность экспорта BPMN диаграмм бизнес-процессов в формат XMI в данный момент отсутствует.

## **Enterprise Architect**

Средство визуального моделирования и проектирования компании Sparx Systems, основанное на OMG UML, охватывающее ключевые аспекты жизненного цикла разрабатываемого приложения от управления требованиями до проектирования, сборки, тестирования и промышленной эксплуатации.

Enterprise Architect поддерживает BPMN 2.0 с возможностью экспорта спроектированных диаграмм в формат XMI.

Средство MDG Integration for Visual Studio, также разработанное Sparx Systems, позволяет интегрировать модели, созданные с помощью Enterprise Architect, в среду разработки Visual Studio.

## **QReal**

QReal – CASE и metaCASE-система с определенным набором графических редакторов и возможностью автоматически генерировать визуальные редакторы по описанию языка, разработанная на кафедре Системного программирования Математико-Механического факультета. На данный момент редактор диаграмм бизнес-процессов в нотации BPMN не реализован.

## Обзор языков моделирования бизнес-процессов

В этой главе приведено краткое описание двух наиболее перспективных стандартов в области моделирования бизнес-процессов - BPMN и BPEL.

### **BPEL**

Язык исполнения бизнес процессов - Business Process Execution Language (также используется полное название - WS-BPEL - Web Services Business Process Execution Language) - является стандартом проектирования и исполнения бизнес процессов. BPEL представляет собой язык на основе XML, позволяющий моделировать взаимодействия между Web-сервисами. Подобное моделирование значимо для успешного управления бизнес процессами и реализации сервис-ориентированной архитектуры (SOA) приложений.

BPEL разрабатывался сотрудниками компаний IBM, Microsoft, BEA Systems, SAP и Siebel на основе языков WSFL (компания IBM) и XLANG (компания Microsoft). В апреле 2003 года BPEL4WS 1.1 была передана международной организации OASIS (Organization for the Advancement of Structured Information Standards - Организация по продвижению стандартов в области структурированной информации); в 2004 году был стандартизирован (спецификацию назвали WS-BPEL 2.0).

Для того чтобы упростить процесс моделирования, BPEL использует простые и понятные команды, которые выполняют сложные функции. BPEL удобен для восприятия, что позволяет бизнес-пользователям и разработчикам работать совместно.

Одним из основных использований BPEL является моделирование взаимодействий Web-сервисов в распределенной системе. BPEL учитывает

сложные оркестровки нескольких сервисных приложений через один контроллер.

WS-BPEL основывается на нескольких спецификациях, таких как SOAP <sup>1</sup>, WSDL <sup>2</sup>, и XML Schema <sup>3</sup>. Среди них WSDL является, пожалуй, самой важной спецификацией. WSDL - то, что делает возможным использование сервиса в составных сервисах, основанных на WS-BPEL. WS-BPEL позволяет определять бизнес-процессы, взаимодействующие с кооперирующими сервисами по WSDL описаниям.

BPEL часто ассоциируют с BPMN, который также стремится упростить процесс моделирования управления бизнес-процессами. В отличие от BPEL, BPMN не является исполняемым, поэтому он в основном используется для планирования и проектирования. BPMN, тем не менее, имеет визуальные компоненты, делающие его более понятным для бизнес-пользователей, не знакомых с программированием. Многие компании разработали свои собственные визуальные нотации для BPEL, чтобы упростить язык.

За последние несколько лет популярность BPMN и BPEL возросла, так как каждый из них стремится упростить управление бизнес-процессами и поддерживает взаимодействие между бизнес-пользователями и разработчиками. Правда преобразование одного из них в другой остается нерешенной задачей [4].

## **BPMN**

Business Process Model and Notation - международный стандарт моделирования, разработанный организацией BPMI <sup>4</sup>; в данный момент

---

<sup>1</sup> Simple Object Access Protocol - протокол обмена структурированными (в формате XML) сообщениями в распределенной вычислительной среде

<sup>2</sup> Web Services Description Language - основанный на XML язык описания интерфейсов, используемый для описания предоставляемой Web-сервисом функциональности

<sup>3</sup> XML Schema - язык описания структуры XML-документа

<sup>4</sup> Business Process Management Initiative

поддерживается рабочей группой OMG <sup>5</sup>. Суть использования BPMN при моделировании бизнес процессов схожа с сутью UML при объектно-ориентированном проектировании и анализе, и заключается в определении наилучших идей в существующих подходах и их объединение в новый, широко используемый язык. Среди предшественников BPMN есть языки моделирования процессов, основанные на графах и сетях Петри, такие, например, как диаграммы активности UML и управляемые событиями цепочки событий (EPC).

Эти языки моделирования фокусируются на разных уровнях абстракции (от бизнес уровня до более технических уровней), в то время как BPMN стремится к поддержке полного спектра уровней абстракции, от бизнес уровня до уровня технической реализации. Эта задача сформулирована в документации стандарта: “Основной целью языка BPMN является обеспечение абсолютно доступной нотацией для описания бизнес процессов всех бизнес-пользователей: от бизнес-аналитиков, создающих схемы процессов, и разработчиков, ответственных за внедрение технологий выполнения бизнес процессов, до руководителей и обычных пользователей, управляющих этими бизнес процессами и отслеживающих их выполнение. Таким образом, BPMN нацелен на устранение расхождения между моделями бизнес-процессов и их реализацией.” [5].

Другой, не менее важной целью разработки BPMN, явилось то, что языки XML (например, WS-BPEL), разработанные для исполнения бизнес-процессов, теперь могут быть визуализированы в графической нотации и тем самым становятся понятными обычным бизнес-пользователям [5].

---

<sup>5</sup> Object Management Group

## Описание решения

### Выбор редактора диаграмм

Выбор Enterprise Architect в качестве редактора диаграмм обусловлен тем, что это средство удовлетворяет сформулированным критериям (связь диаграмм бизнес-процессов и диаграмм классов и возможность экспорта в xml-формат), а также возможностью импортировать спроектированные классы в рабочую среду и возможность подключения спроектированных моделей в среду разработки Visual Studio 2010.

Каждый объект данных, участвующий в бизнес-процессе, и, соответственно, отображенный на диаграмме в виде Элемента Данных (Information Item) или Объекта Данных (Data Object), должен быть экземпляром некоторого класса из диаграммы классов UML или быть полем некоторого класса. Средство Enterprise Architect позволяет привязывать элементы данных к конкретным классам из диаграммы классов:

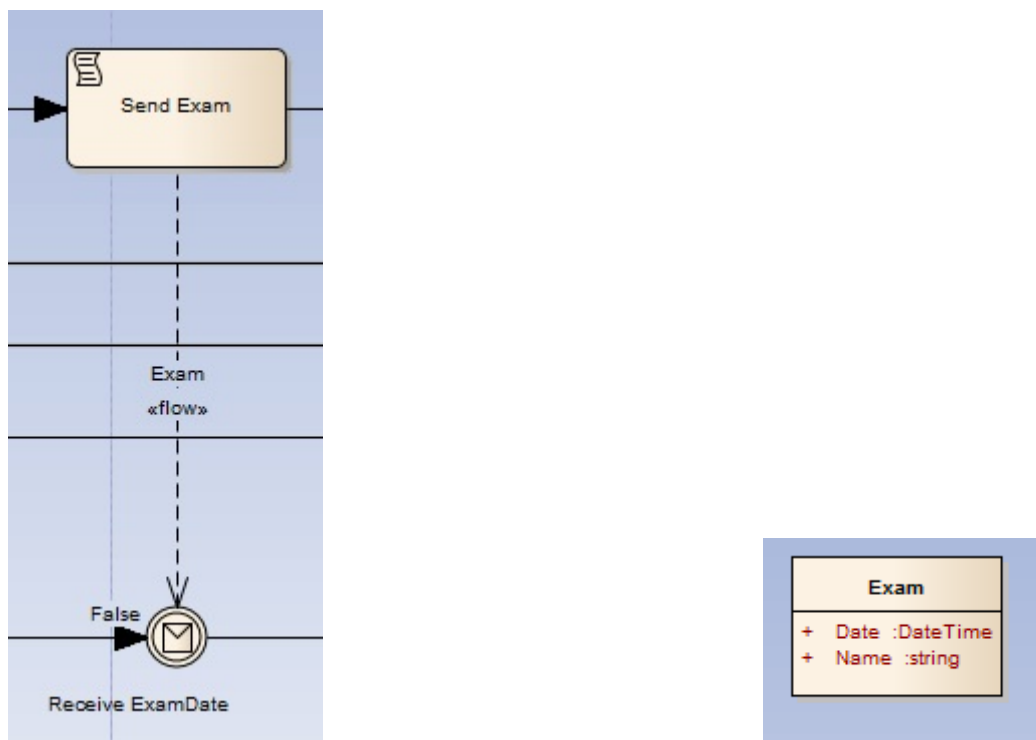


Рисунок 1. Пересылка объекта данных Exam между участниками бизнес-процесса; Рисунок 2. Класс Exam в диаграмме классов

Также средство Enterprise Architect обеспечивает связь с моделями данных, спроектированными в среде Visual Studio . Импортированные модели могут быть привязана к диаграмме бизнес-процессов:

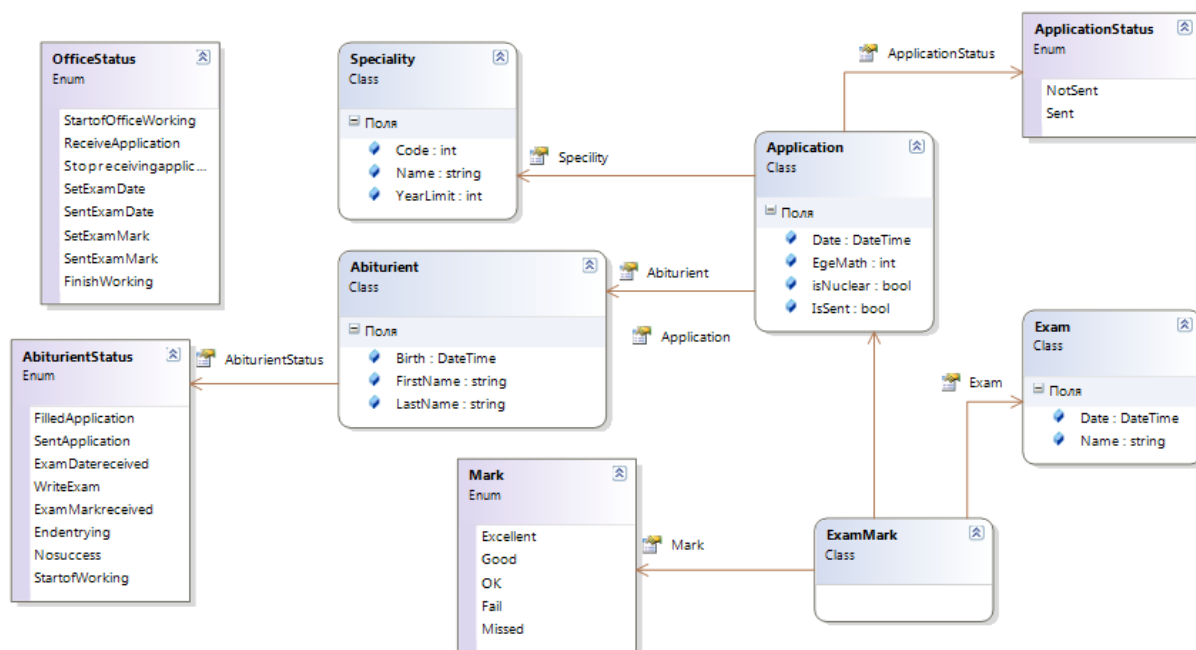


Рисунок 2. Диаграмма классов, нарисованная в Visual Studio

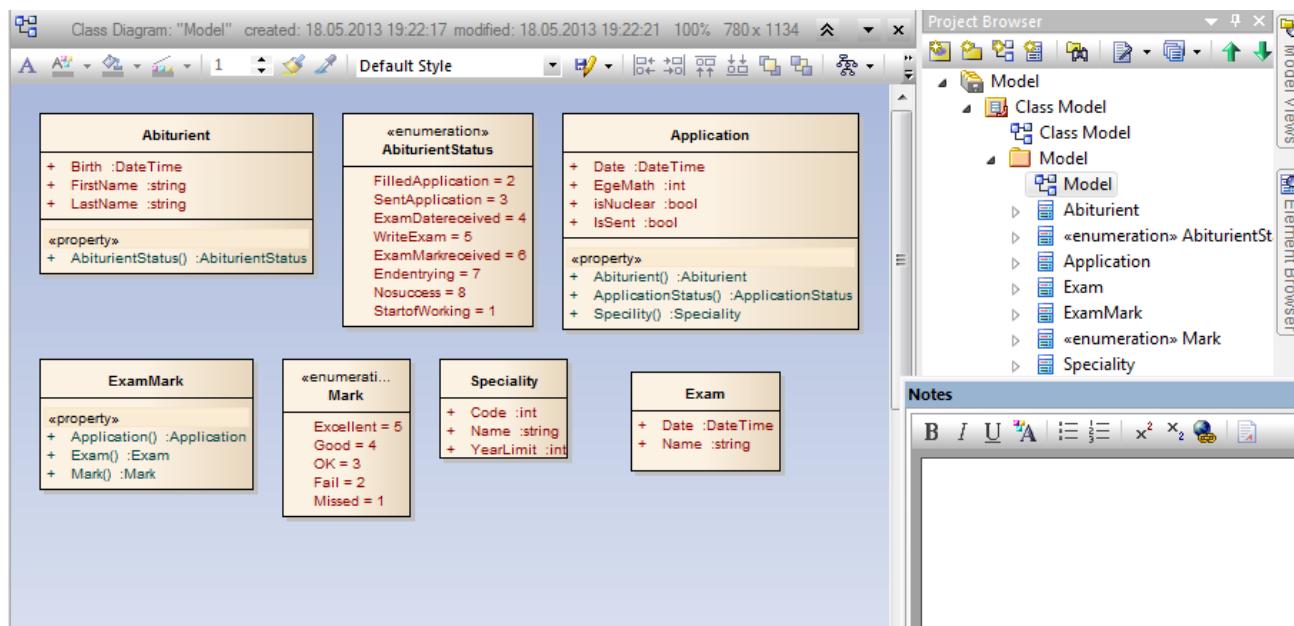


Рисунок 3. Результат загрузки классов, сгенерированных по диаграмме, изображенной на рис.2

## Генерация кода бизнес-процессов

Проведя анализ требований к существующим ИС, разработанным на основе технологии REAL-IT/.NET и кода ручных изменений, внесённых для поддержки динамических процессов, можно сформулировать следующие требования к генераторам кода:

1. Генерация служебных состояний для различных сущностей, участвующих в процессах
2. Генерация статических классов для уникальных объектов
3. Генерация извлекаемых из диаграммы ограничений на доступ к данным в те или иные этапы процессов, а также элементов интерфейса, отвечающих за смену состояний объектов.

## Создание служебных состояний

На BPMN диаграмме отдельно взятые пулы представляют собой различных участников бизнес-процесса (пользователи информационной системы). Такие участники именуется бизнес-сущностями (бизнес-ролями).

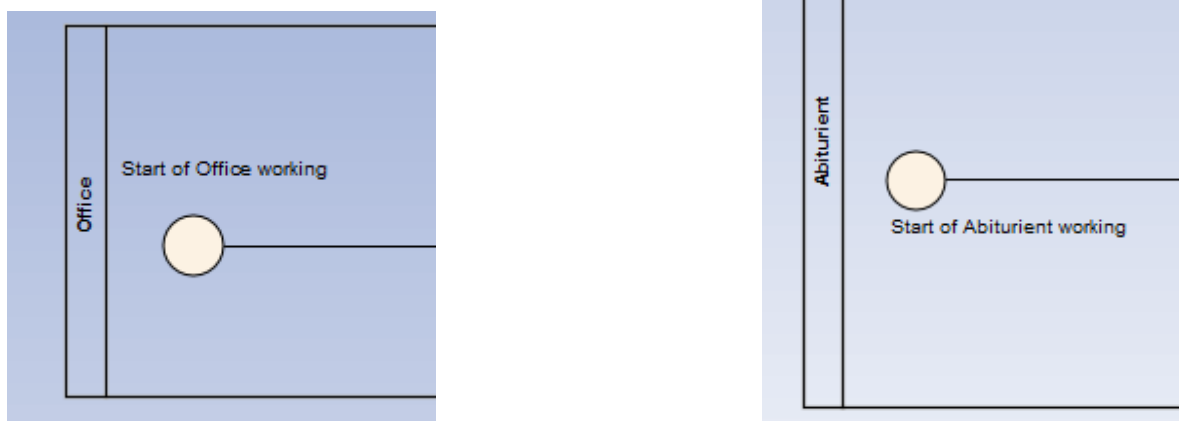


Рисунок 4. Участники бизнес-процесса

Каждая бизнес-сущность должна быть экземпляром некоторого класса из модели данных (в приведенном примере, однако, Приемная комиссия (Office))



не является экземпляром некоторого класса, т.к. Приемная комиссия существует ровно в одном экземпляре).

Было предложено извлекать из спроектированной диаграммы бизнес-процессов так называемые служебные состояния, отвечающие набору статусов, в которых может пребывать каждый из участников бизнес-процесса. В определенный момент времени участник процесса находится в одном из служебных состояний. Каждое служебное состояние соответствует событию (начальному (Start Event), промежуточному (Intermediate Event) или конечному (End Event)) или Пользовательскому заданию (User Task)). Для каждого служебного состояния генерировался соответствующий enumeration, являющийся свойством класса, соответствующей сущности.

### **Работа со статическими классами, сохраняющими состояние уникальных объектов**

Если существует бизнес-сущность, которая является уникальным объектом, то согласно текущим правилам технологии, в модели данных за ненадобностью будет отсутствовать класс, соответствующий этой бизнес-сущности. Но, вследствие необходимости сохранения состояния этого объекта было решено генерировать соответствующий класс.

У этого класса имеется статическое свойство, отвечающее за состояние уникального объекта в данный момент. Соответственно, изменение в ходе работы ИС состояния уникального объекта означает изменение статического свойства. Идентификационный номер текущего состояния уникального объекта хранится в соответствующей таблице в базе данных.

Также, требование поддержки доступа к состояниям уникальных сущностей выявила необходимость генерации классов, не относящихся напрямую к какой-либо форме.

Дополнительно заметим, что в процессе развития ИС встречаются случаи, когда появляется необходимость отмены требования об

уникальности объекта. В случае с использованием статических классов, провести такое изменение становится гораздо легче.

## **Получаемые из диаграммы бизнес-процессов ограничения и действия**

Необходимо сформулировать ограничения, которые будут извлекаться из диаграммы бизнес-процессов и внести необходимые изменения в генератор кода.

Была предложена следующая идея: все изменения раскладываются в некоторый базис, состоящий из элементарных ограничений и элементарных действий. Сформулируем составные элементы этого базиса.

Элементарные ограничения:

1. Ограничения на изменение поля - возможность редактирования поля карточки объекта;
2. Ограничения на создание/редактирование/удаление сущностей - возможность совершать соответствующие операции со списками (включая встроенные списки);
3. Ограничения на статусы (служебные состояния) – из текущего состояния участник бизнес-процесса может перейти лишь в состояния, связанные с нынешним на диаграмме (таким образом, учитывается порядок состояний).

Элементарные действия:

1. Изменение значения поля объекта в зависимости от текущего состояния;
2. Включение/выключение элементарных ограничений.

Действия соответствует кнопке на карточке бизнес-сущности, соответствующие Пользовательским заданиям (User Task).

Каждая такая кнопка:

1. Имеет имя, соответствующее имени задания, по которому оно сгенерировано;
2. Становится активной, когда бизнес-сущность переходит в состояние, предшествующее состоянию, сгенерированному по заданию, по которому эта кнопка и сгенерирована. Если на диаграмме задание соединено информационным потоком с промежуточным событием-сообщением из другого пула, то необходимо аналогичное условие на состояние этой бизнес-сущности;

При нажатии происходят следующие события:

1. Если задание, соответствующее кнопке, соединено информационным потоком с промежуточным событием-сообщением из другого пула, то происходит смена состояний бизнес-сущностей, накладываются ограничения на редактирование/удаление объектов данных, ассоциированных с информационным потоком.
2. В противном случае, происходит смена состояния бизнес-сущности.

После этого кнопка становится неактивной.

Для поддержки каждого из перечисленных выше элементарных операций был изменён шаблон генерируемой формы и добавлена соответствующая логика в код генератора.

## **Изменение библиотек поддержки исполнения**

Была выделена функциональность, общая для всех приложений, и соответствующие изменения были внесены в библиотеки поддержки исполнения, что позволило сократить количество сгенерированного кода.

Основные изменения:

1. Работа с состояниями,
2. Обеспечение доступа к полям БД, сохранение значений статических объектов,
3. Ограничение доступа к полям форм.

## Апробация

Рассмотрим пример ИС “Поступление в университет”.

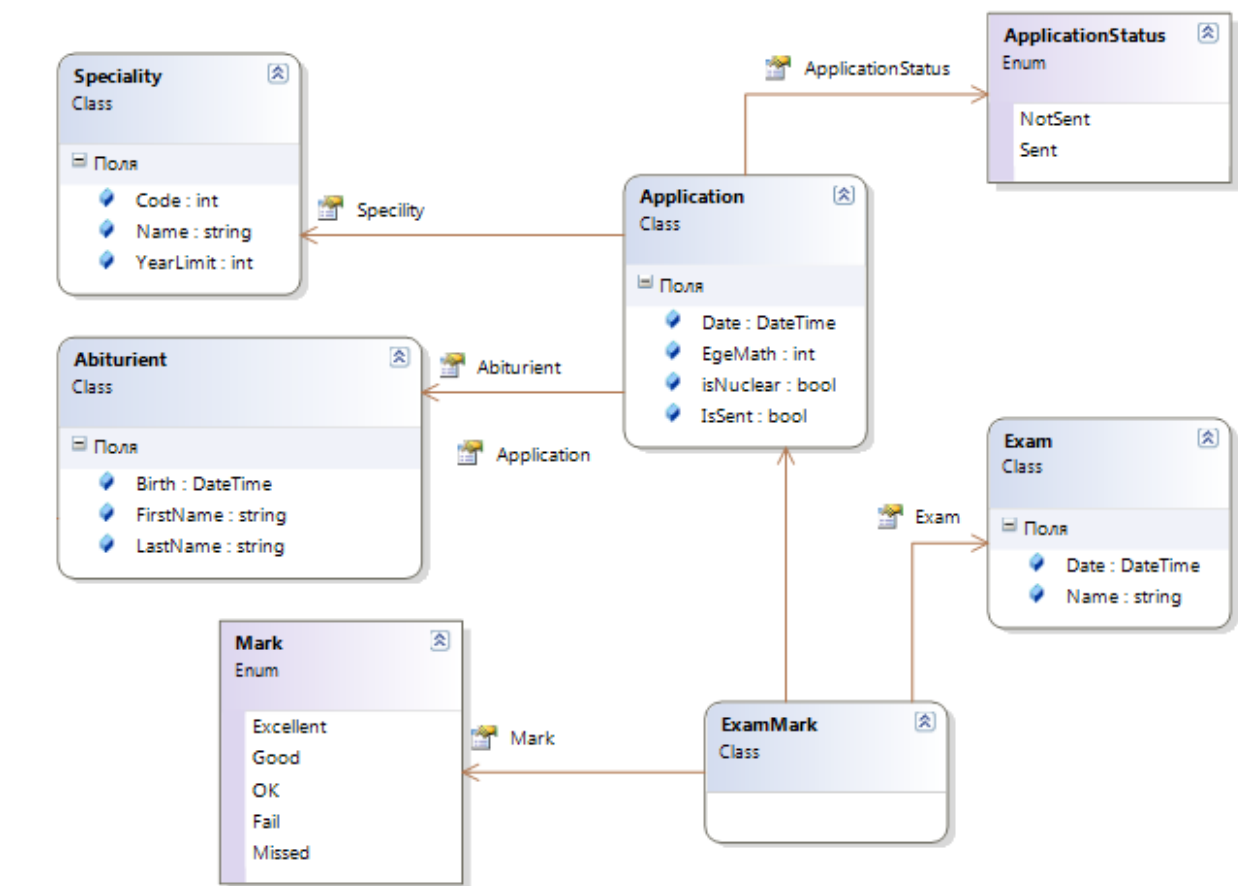


Рисунок 5. Спроектированная диаграмма классов

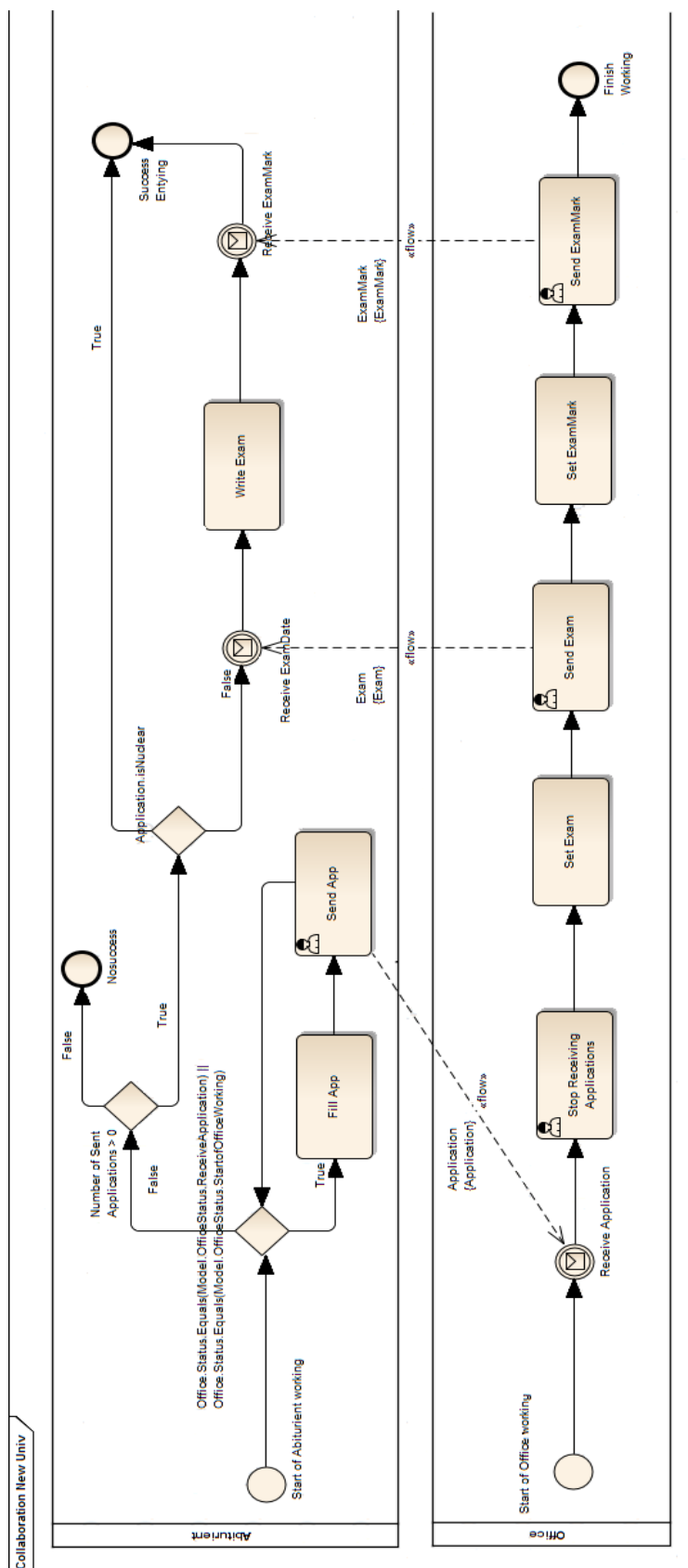
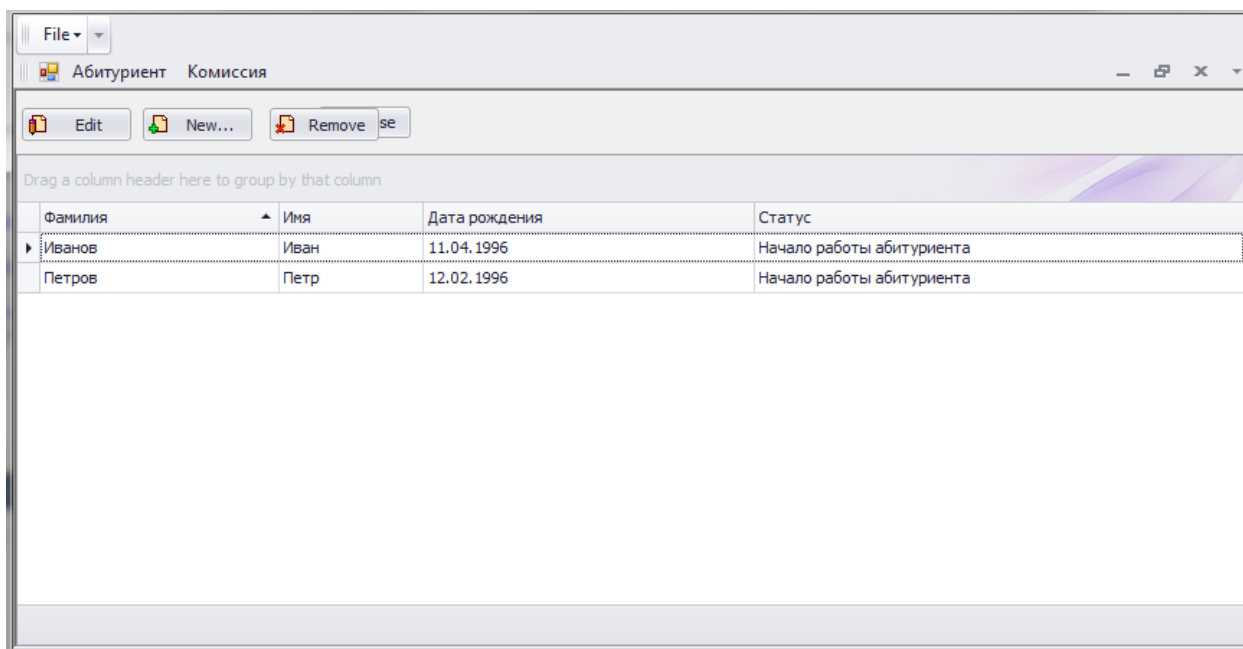


Рисунок 6. Диаграмма бизнес-процесса, спроектированная в Enterprise Architect 10



File ▾

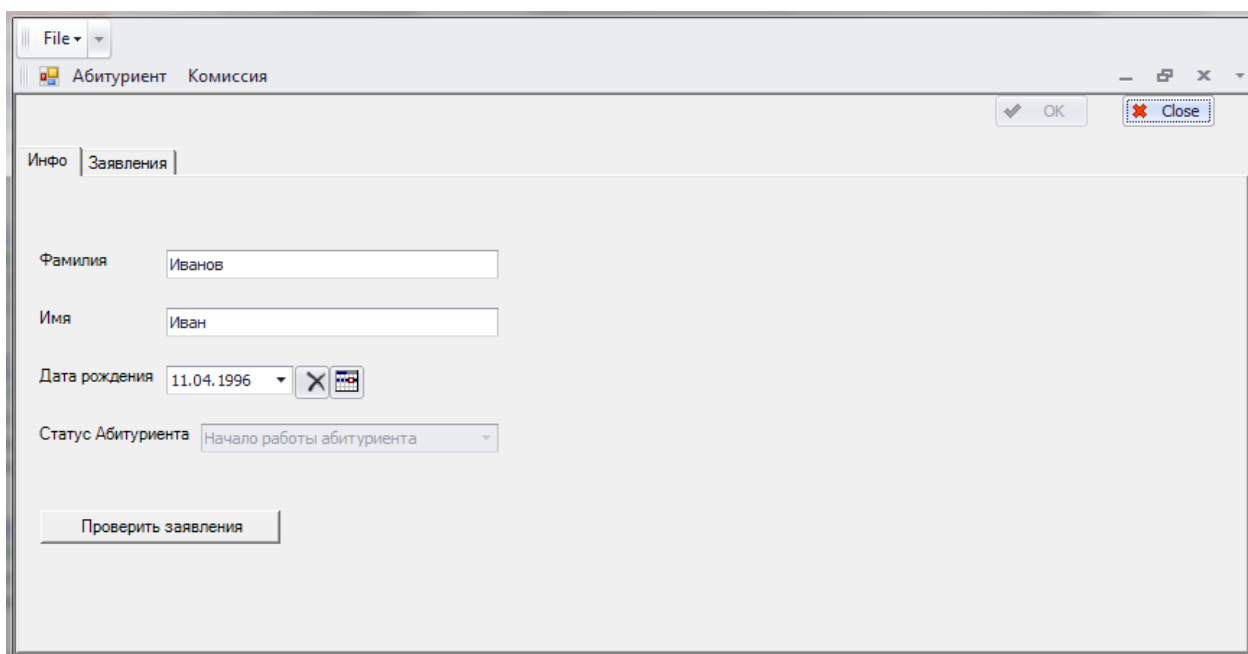
Абитуриент Комиссия

Edit New... Remove se

Drag a column header here to group by that column

Фамилия	Имя	Дата рождения	Статус
Иванов	Иван	11.04.1996	Начало работы абитуриента
Петров	Петр	12.02.1996	Начало работы абитуриента

Рисунок 7. Список абитуриентов в начале работы



File ▾

Абитуриент Комиссия

OK Close

Инфо Заявления

Фамилия Иванов

Имя Иван

Дата рождения 11.04.1996

Статус Абитуриента Начало работы абитуриента

Проверить заявления

Рисунок 8. Карточка абитуриента. Невозможно поменять статус абитуриента вручную

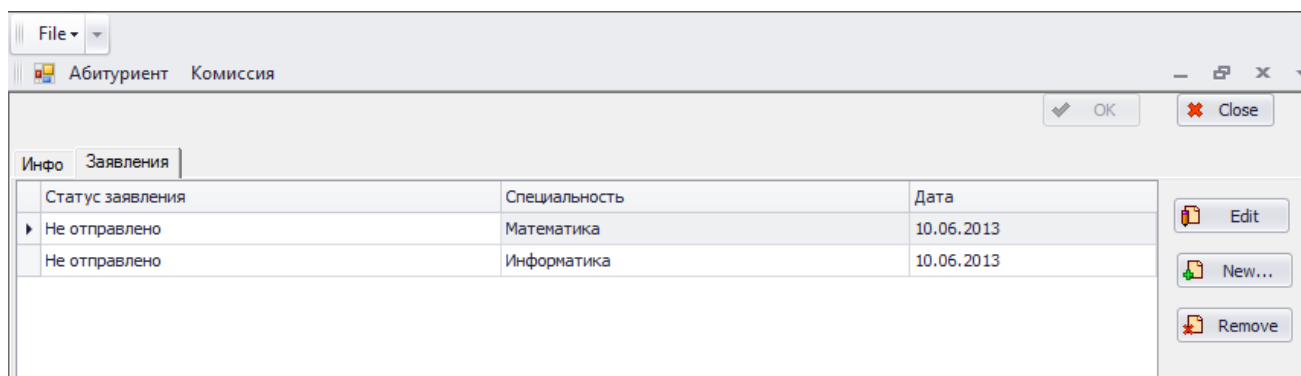


Рисунок 9. Список неотправленных заявлений абитуриента

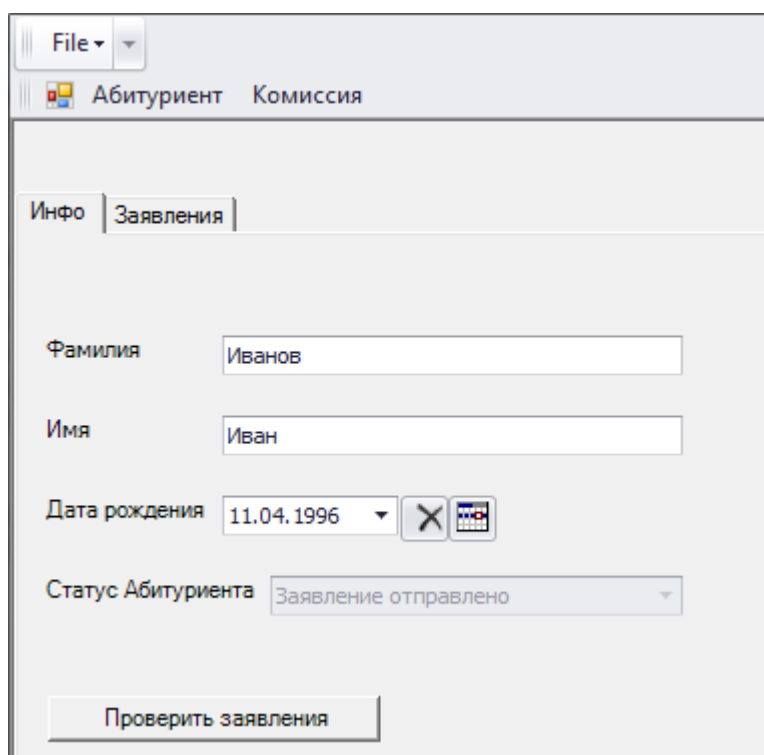


Рисунок 10. Посылка заявлений. Статус абитуриента изменился

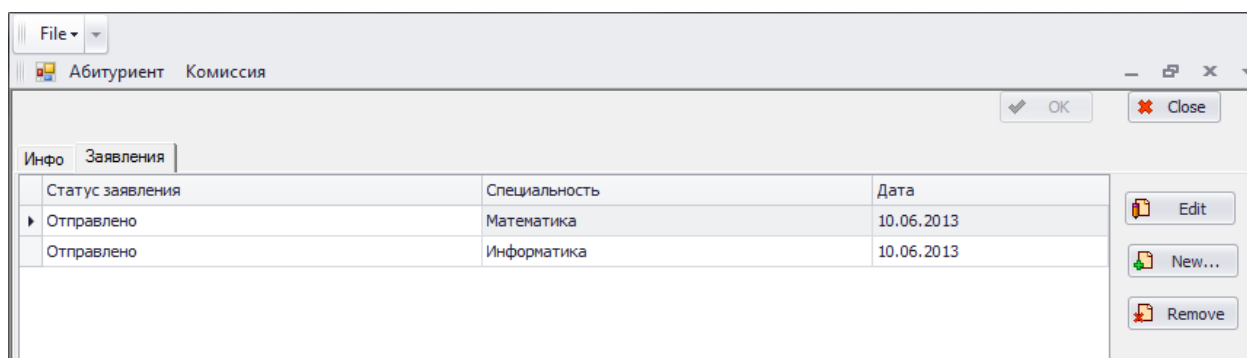


Рисунок 11. Заявления действительно отправились

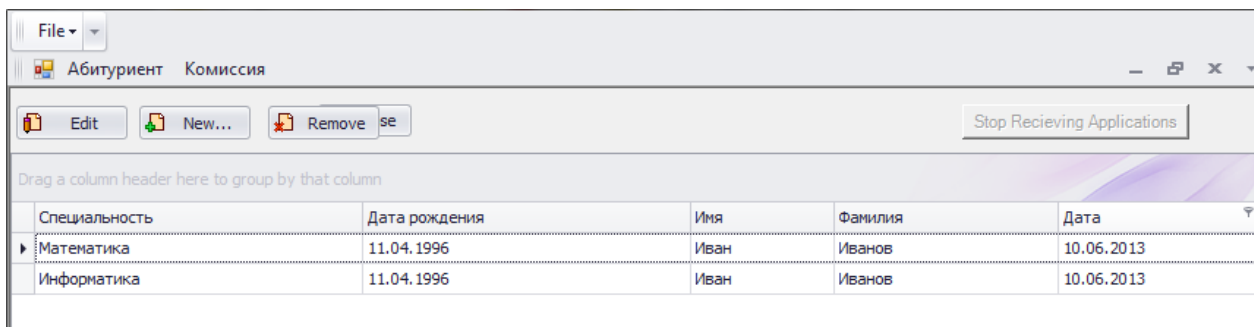


Рисунок 12. Приемная комиссия останавливает прием

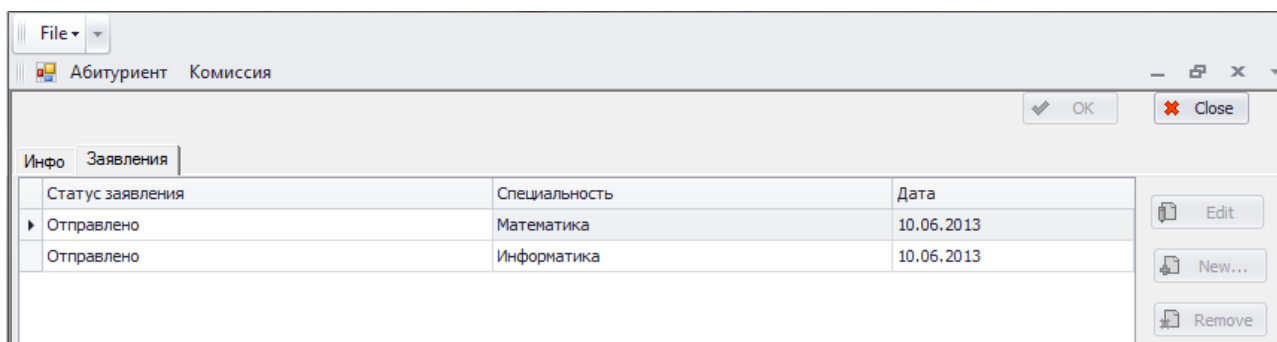


Рисунок 13. Невозможно создать новые заявления

Фамилия:

Имя:

Дата рождения:

Статус Абитуриента:

Рисунок 14. Абитуриент, не приславший заявлений, имеет статус "Неудача", соответствующий Конечному событию на диаграмме бизнес-процессов



## **Заключение**

В рамках данной дипломной работы выполнены следующие задачи:

1. Реализованы общие для всех приложений функции и внесены соответствующие изменения в библиотеки поддержки исполнения;
2. Внесены изменения в редакторы экранных форм и базы данных, генерирующие ограничения, полученные из диаграмм бизнес-процессов, спроектированных в выбранном редакторе диаграмм;
3. Полученные генераторы апробированы на спроектированном примере информационной системы “Поступление в университет”.

## **Направления дальнейшей работы**

1. Разграничение прав доступа к ресурсам для разных категорий пользователей информационной системы: на данный момент порядок и условия доступа пользователя к данным, хранящимся в информационной системе, никак не регламентируется. Планируется ввести процедуру идентификации и аутентификации.
2. Работа с временными событиями: При моделировании бизнес-процесса зачастую встает необходимость задавать выполнение определенных действий по прошествии некоторого периода времени или по достижении некоторой даты. В нотации BPMN 2.0 для подобных целей используются специальные события-таймеры (Timer Events).
3. Работа с QReal: Планируется создание с помощью metaCase-системы QReal редактора диаграмм и среды исполнения для визуального языка на основе BPMN.

## Список литературы

- [1] Иванов А.Н. Автоматизированная генерация информационных систем, ориентированных на данные, 2005 г.
- [2] Нестеров А.В. Перенос технологии REAL-IT на платформу Microsoft .NET, 2007 г.
- [3] Терехов А.Н., Романовский К.Ю., Кознов Д.В., Долгов П.С., Иванов А.Н. Real: методология и CASE-средство для разработки систем реального времени и информационных систем. Программирование. N.5, М. Наука 1999. С. 44-51.
- [4] Введение в BPEL <http://searchsoa.techtarget.com/tutorial/BPEL-tutorial>
- [5] Спецификация BPMN 2.0 <http://www.elma-bpm.ru/bpmn2/>
- [6] Баланс между оркестровкой и межпроцессорным взаимодействием в BPMN <http://mainthing.ru/ru/item/439/>
- [7] Белайчук А. Избранные паттерны BPM, “Открытые системы”, № 01, 2009 г.
- [8] Ferdian T. A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes, 2001 г.
- [9] Anni Tsai EPC Workflow Model to WIFA Model Conversion, 2006 г., стр. 2758-2763
- [10] van Hee K., Oanea O., Sidorova N. Colored Petri Nets to Verify Extended Event-Driven Process Chains, 2005 г.
- [11] Кулябов Д., Королькова А. Введение в формальные методы описания бизнес-процессов, 2008 г.

- [12] IDEF <http://searchsoa.techtarget.com/definition/IDEF>
- [13] Волков О. Стандарты и методологии моделирования бизнес-процессов, 2005 г.
- [14] IDEF0 <http://www.edrawsoft.com/IDEF0-flowcharts.php>
- [15] Вендров А. CASE-технологии. Современные методы и средства проектирования информационных систем, 2003 г.
- [16] Норенков И. Подходы к проектированию автоматизированных систем, 2005 г.