

Дипломная работа на тему
«Интеграция «ИС Деканат» и
«1с Университет» со СМЭВ»

Выполнил: Толпыго Е.О.

Научный руководитель: Ивашко А.Г.

Цель взаимодействия

- Передача данных о студенте ТюмГУ (является студентом или нет) МФЦ

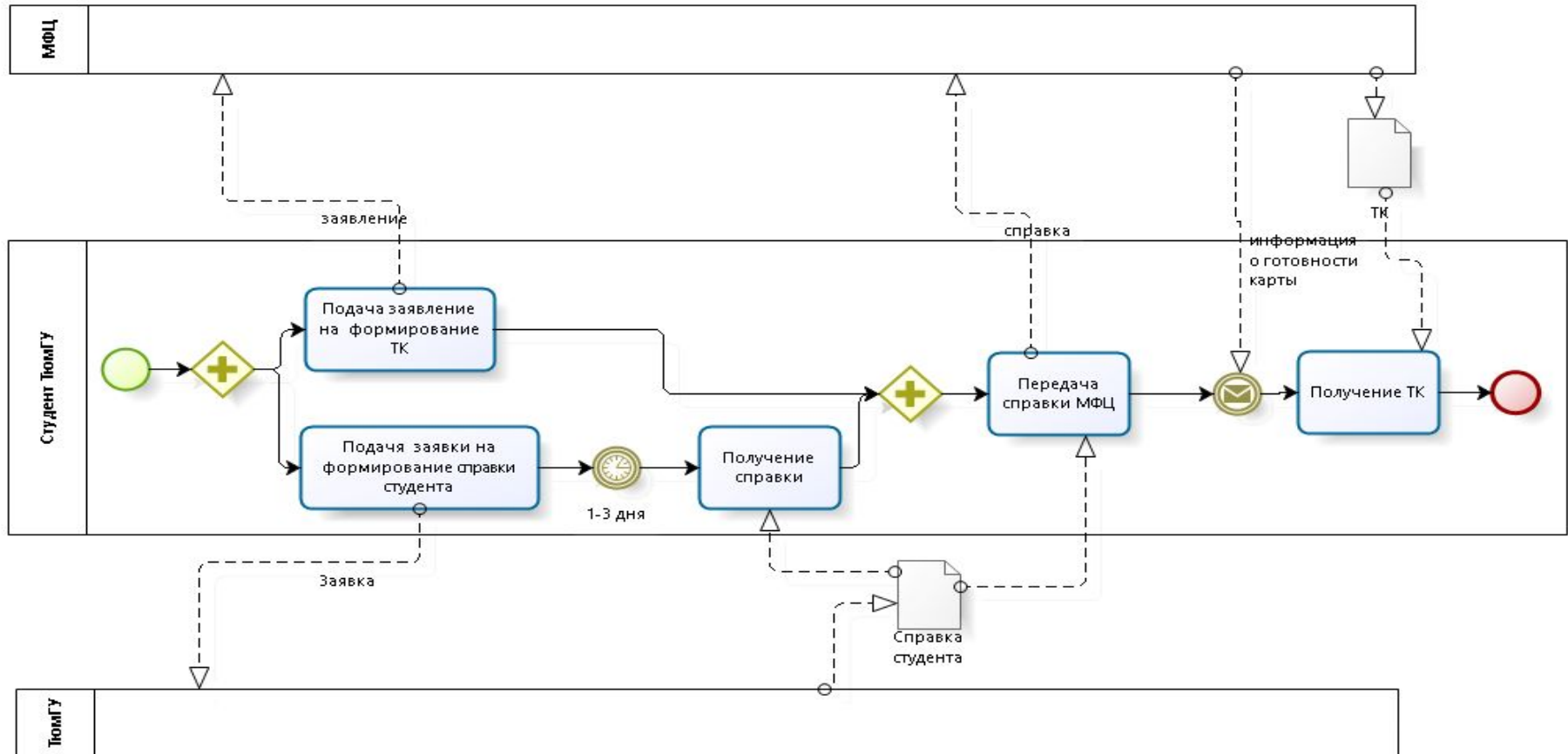
Передаваемые данные МФЦ

- Имя студента
- Фамилия студента
- Отчество студента
- Дата рождения студента
- Серия и номер документа подтверждающий личность
- Дата выдачи документа подтверждающий личность

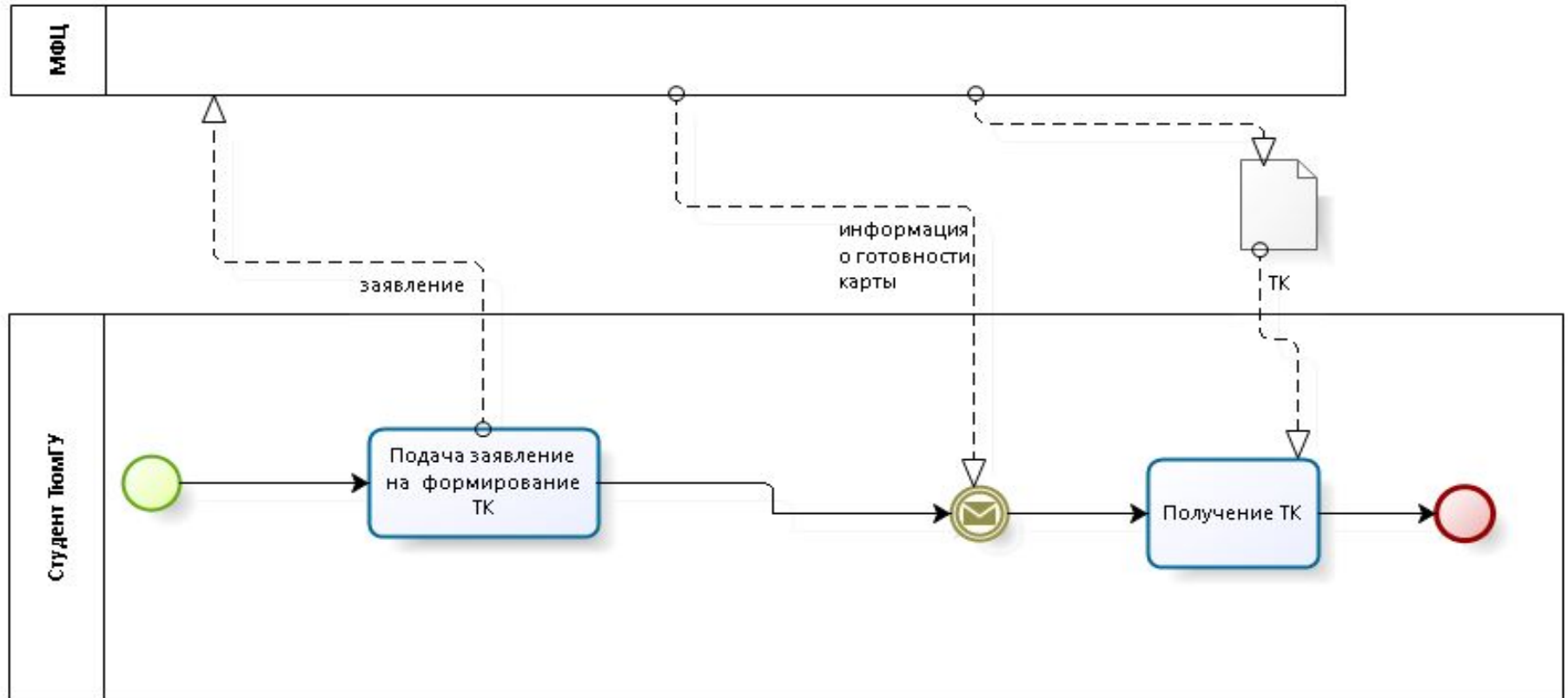
Передаваемые данные ТюмГУ

- Имя студента
- Фамилия студента
- Отчество студента
- Дата рождения
- Номер группы
- Факультет
- Дата отчисления (если был отчислен)
- Статус

Оформление транспортной карты до интеграции



Оформление транспортной карты после интеграции



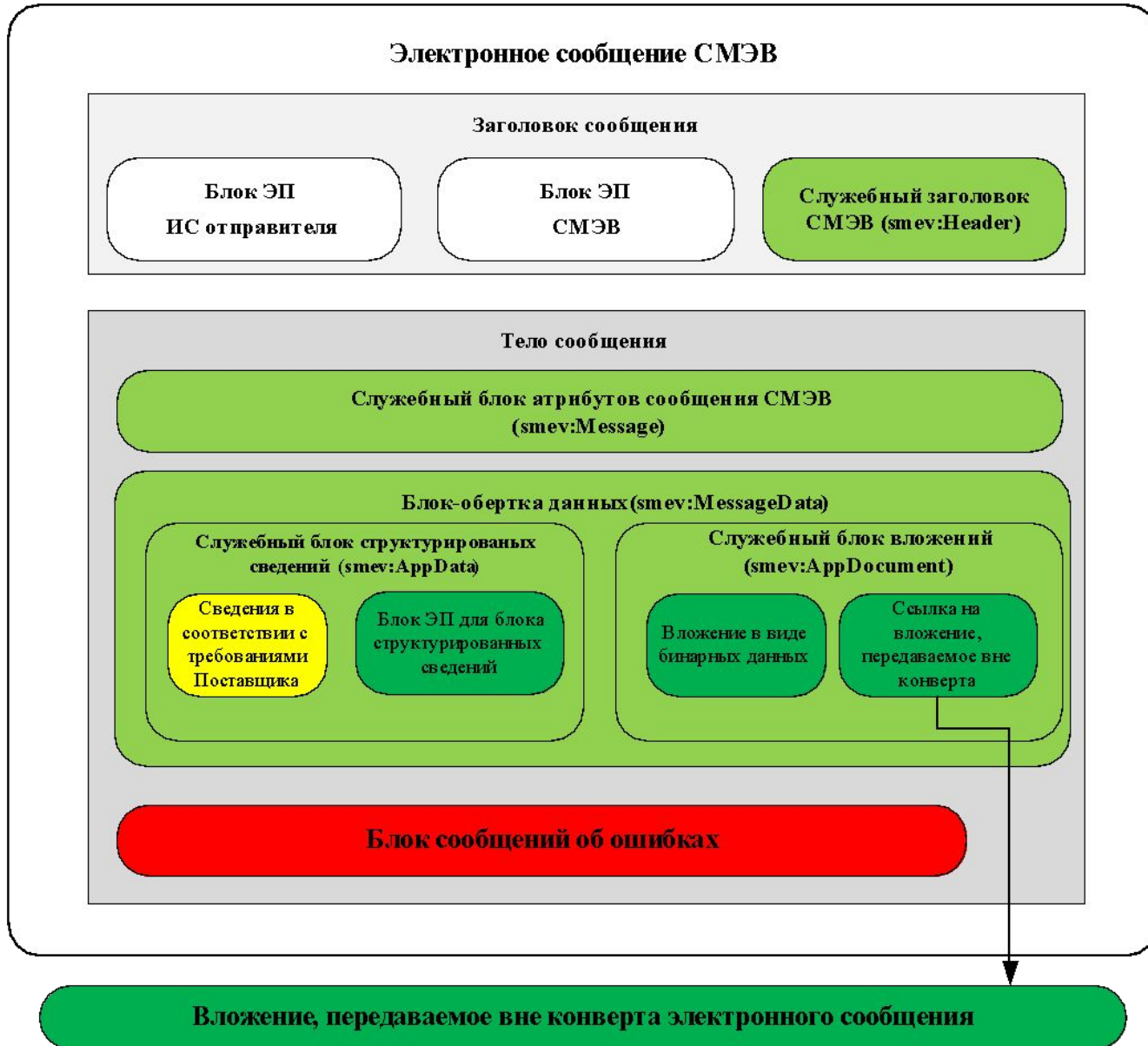
Способы интеграции систем

- Веб – сервисы
- Технология обмена файлами
- Модель сервер-сервер и применением шлюзов баз данных

Взаимодействие через СМЭВ



Структура СМЭВ сообщения



Цель разработки

- Уменьшение временных затрат на получение сведений о студенте за счет автоматизации процесса
- Исключение процента ошибок
- Исключение коррупционных действий

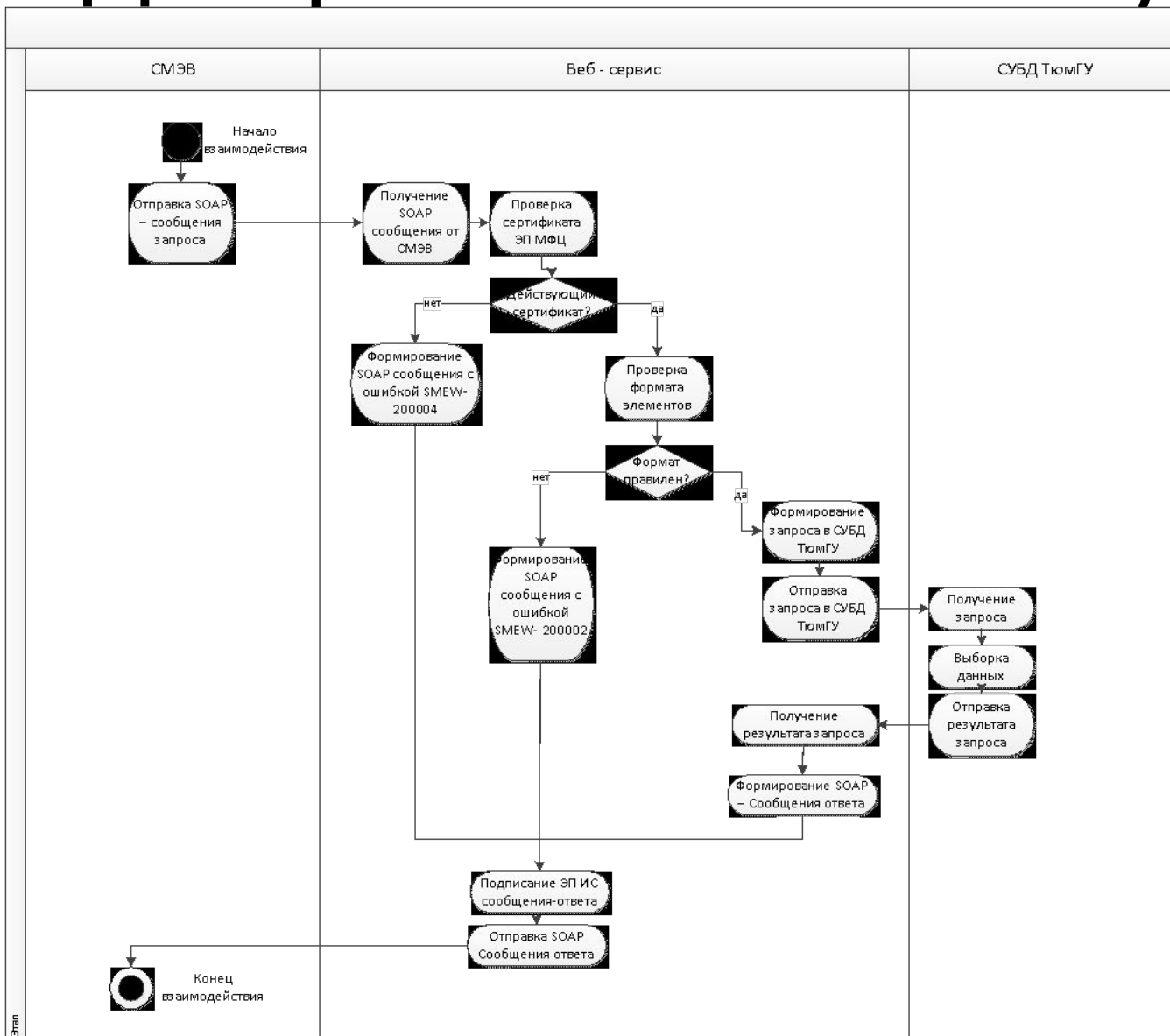
Задачи

- Изучить предметную область
- Найти проблему и способы ее решения
- Сделать диаграмму в нотации UML Activity, которая будет описывать поведение веб – сервиса
- Создать веб – сервис
- Тестирование
- Составить спецификацию веб – сервиса и передать ее МФЦ
- Обеспечить работу с «1С Университет» после перехода
- Составить документацию по проделанной работе

Функциональные требования

- Проверка сертификата электронной подписи
- Запрос на получение данных в «Деканат»
- Запрос на получение данных в «1С Университет»
- Формирование и отправка сообщения-ответа
- Формирование электронной подписи

Диаграмма UML Activity



Способы передачи данных WCF

- Использование контрактов данных (DataContractSerializer);
- Использование класса XmlSerializer
- Использование контрактов сообщений
- Использование класса сообщений (Message)
- Изменение структуры сообщения на прямую (существует несколько способов)

Интерфейс веб - сервиса

```
[ServiceContract (ProtectionLevel = ProtectionLevel.Sign, Namespace = "http://UTMNTtoMFC.ru/")]
```

ссылка 1

```
public interface IService1
```

```
{
```

```
    [XmlSerializerFormat]
```

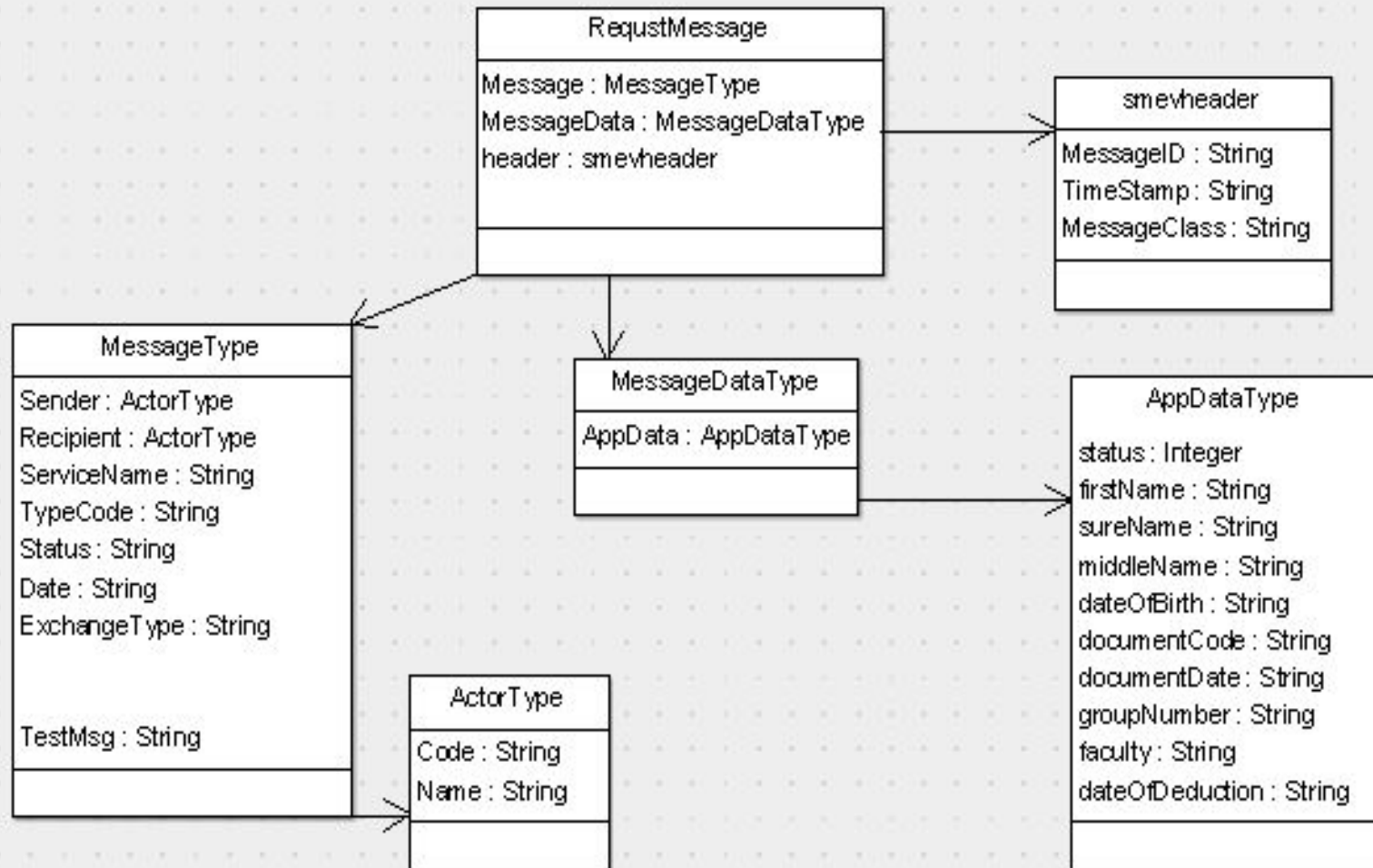
```
    [OperationContract]
```

ссылка 1

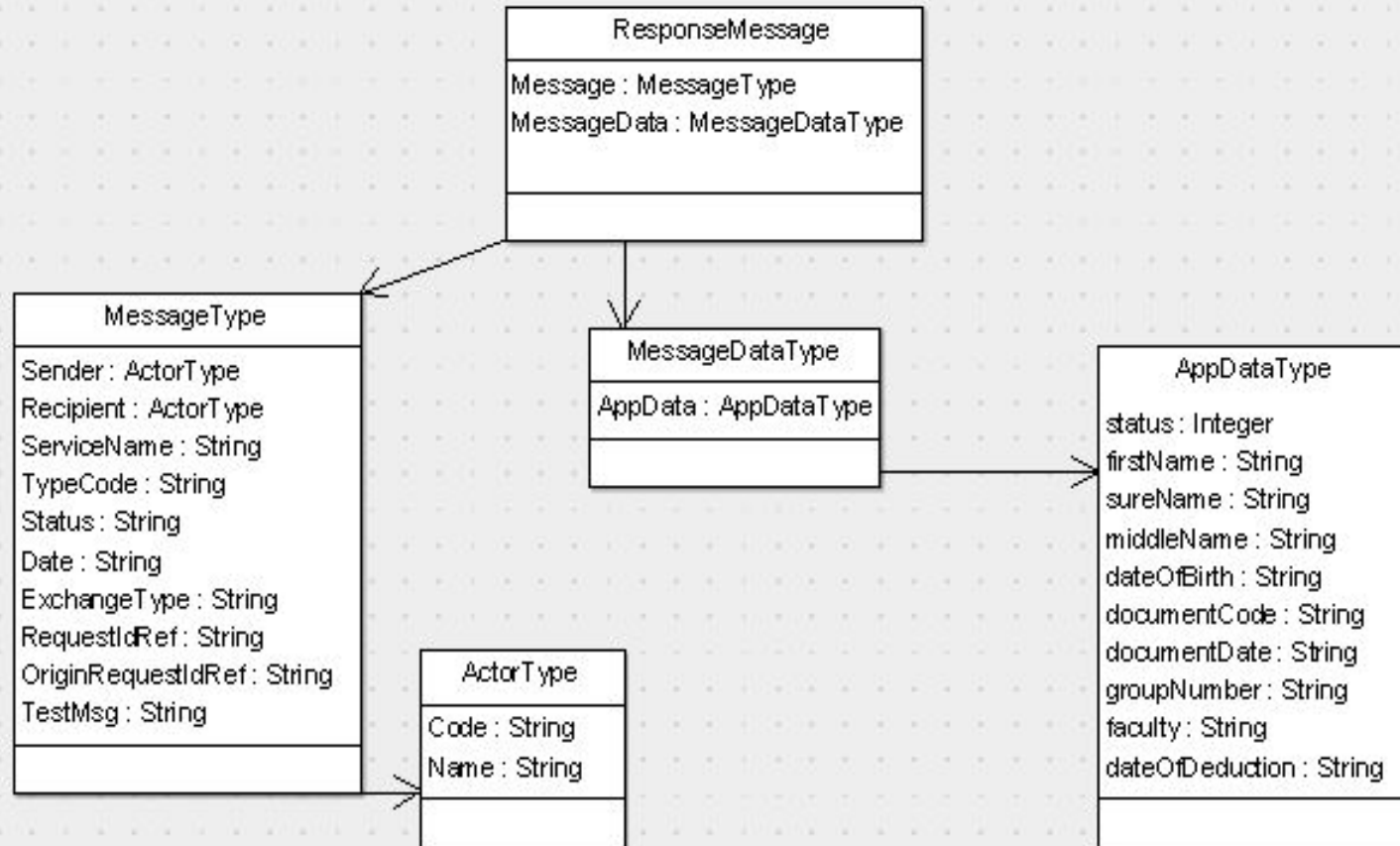
```
    ResponseMessage confirmStudentStatus(RequestMessage request);
```

```
}
```

Диаграмма класса «RequestMessage»



«ResponseMessage»



Пример использования XmlSerializer

Код

```
[XmlRoot]
ссылка 4
public class ActorType
{
    [XmlElement]
    public string Code;
    [XmlElement]
    public string Name;
}
```

XML

```
<Actor>
    <Code/>
    <Name/>
</Actor/>
```

Префикс пространства имен

```
[XmlNamespaceDeclarations]
```

```
ссылка 0
```

```
public XmlSerializerNamespaces xmlsn  
{  
    get  
    {  
        XmlSerializerNamespaces xsn = new XmlSerializerNamespaces();  
        xsn.Add("smev", "http://smev.gosuslugi.ru/rev120315");  
        xsn.Add("utmnmfc", "http://UTMntoMFC.ru");  
        return xsn;  
    }  
    set  
    {  


---



---

  
    }  
}
```

Пример использования контракта сообщений

```
[MessageContract]
```

```
ссылка 2
```

```
public class RequestMessage
```

```
{
```

```
    [MessageBodyMember(Namespace = "http://smev.gosuslugi.ru/rev120315")]
```

```
    public MessageType Message = new MessageType();
```

```
    [MessageBodyMember(Namespace = "http://smev.gosuslugi.ru/rev120315")]
```

```
    public MessageDataType MessageData = new MessageDataType();
```

```
    [MessageHeader(Namespace = "http://smev.gosuslugi.ru/rev120315")]
```

```
    public smevheader header = new smevheader();
```

```
}
```

Взаимодействие с ИС «Деканат»

```
string connStr = @"Data Source=HOME;  
Initial Catalog=test;  
Integrated Security=True";  
  
SqlConnection conn = new SqlConnection(connStr);  
try  
{  
    conn.Open();  
}  
catch (SqlException se)  
{  
    this.status = 5;  
}  
using (SqlCommand cm = new SqlCommand("getstudentMFC2", conn))  
{  
    cm.CommandType = CommandType.StoredProcedure;  
  
    SqlParameter[] param = new SqlParameter[10];
```

```
param[5].ParameterName = "@DocumentDate";  
param[5].Direction = ParameterDirection.Input;  
param[5].SqlDbType = SqlDbType.VarChar;  
param[5].Size = 50;  
param[5].Value = documentDate;
```

```
//выходные  
param[6].ParameterName = "@Status";  
param[6].Direction = ParameterDirection.Output;  
param[6].SqlDbType = SqlDbType.Int;
```

```
cm.Parameters.AddRange(param);
```

```
cm.ExecuteNonQuery();
```

```
this.status = Convert.ToInt32(cm.Parameters["@Status"].Value);
```

```
this.groupNumber = Convert.ToString(cm.Parameters["@GroupNumber"].Value);
```

```
this.faculty = Convert.ToString(cm.Parameters["@Faculty"].Value);
```

```
this.dateOfDeduction = Convert.ToString(cm.Parameters["@DateOfDeduction"].Value);
```

```
conn.Close();
```

Взаимодействие с ИС «1с Университет»

```
public void University(string name, string lastName, string middlename, string dateOfBirth, string documentCode, string docu
{
    var client1c = new ServiceUniversity();
    Request RequestMessage = new Request();
    RequestMessage.firstName = name;
    RequestMessage.lastName = lastName;
    RequestMessage.middleName = middlename;
    RequestMessage.dateoOfBirth = dateOfBirth;
    RequestMessage.documentCode = documentCode;
    RequestMessage.documentDate = documentDate;

    Response ResponseMessage;
    ResponseMessage = client1c.GetStudent(RequestMessage);

    this.groupNumber = ResponseMessage.groupNumber;
    this.status = ResponseMessage.status;
    this.faculty = ResponseMessage.faculty;
    this.dateOfDeduction = ResponseMessage.dateOfDeduction;
}
```


Настройки конечной точки

```
<services>  
  <service name="ConfirmationOfUTMNSTudentStatus.Service1">  
    <endpoint address="" binding="customBinding"  
      bindingConfiguration="smevBinding"  
      contract="ConfirmationOfUTMNSTudentStatus.IService1"/>  
  </service>  
</services>
```

Настройки соединения

```
<bindings>
  <customBinding>
    <binding name="smevBinding">
      <customMessageEncoding logPath="C:\\Games\\TestServiceLog\\"/>
      <security allowSerializedSigningTokenOnReply="true"
        authenticationMode="MutualCertificateDuplex"
        defaultAlgorithmSuite="BasicGostObsolete"
        includeTimestamp="false"
        requireDerivedKeys="true"
        keyEntropyMode="CombinedEntropy"
        messageProtectionOrder="SignBeforeEncrypt"
        messageSecurityVersion="WSecurity10WSTrustFebruary2005WSSecureConversationFebruary2005WSecurityPolicy11BasicSecurityProfile10"
        requireSecurityContextCancellation="false"
        requireSignatureConfirmation="false"
        securityHeaderLayout="Strict">
        <issuedTokenParameters/>
        <localClientSettings detectReplays="false"/>
        <localServiceSettings detectReplays="false"/>
      </security>
      <httpTransport manualAddressing="false"/>
    </binding>
  </customBinding>
</bindings>
```

Настройка сертификата

```
<serviceCredentials>  
  <serviceCertificate findValue="2c 2d 47 17 0b ed 61 74 f5 7e 09 8f 40 b0 97 41 31 50 87 5b"  
    storeLocation="CurrentUser"  
    storeName="My"  
    x509FindType="FindByThumbprint"/>  
  <clientCertificate>  
    <authentication certificateValidationMode="None" revocationMode="NoCheck"></authentication>  
  </clientCertificate>  
</serviceCredentials>
```

Тестирование сообщения

Электронное сообщение для проверки *

```
<utmnmfc:middleName>Олегович</utmnmfc:middleName>  
<utmnmfc:dateOfBirth>02.06.1993</utmnmfc:dateOfBirth><utmnmfc:documentCode />  
<utmnmfc:documentDate />  
<utmnmfc:groupName>25ИСИТ119</utmnmfc:groupName>  
<utmnmfc:faculty>мат.фак</utmnmfc:faculty>  
<utmnmfc:dateOfDeduction /></smev:AppData>  
</smev:MessageData></ResponseMessage>  
</s:Body></s:Envelope>
```

Тип сообщения *

Версия методических рекомендаций *

Тип подписи *

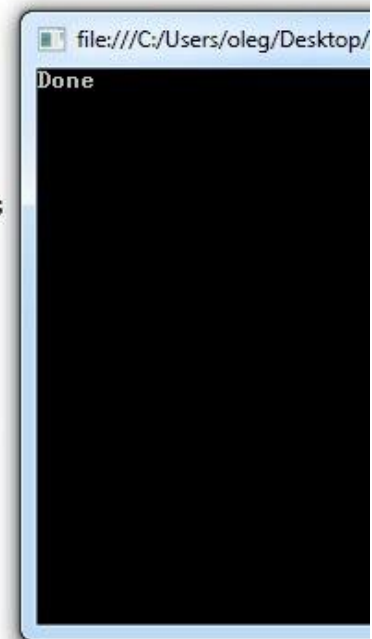
Электронное сообщение соответствует методическим рекомендациям по разработке веб-сервисов версии 2.6.0

Электронная подпись ЭП-ОВ сообщения корректна

Тестирование веб - сервиса

```
RequestMessage asd = new RequestMessage();
asd.Message = new MessageType();
asd.MessageData = new MessageDataType();
asd.MessageData.AppData = new AppDataType();
asd.Message.Sender = new ActorType();
asd.Message.Recipient = new ActorType();
asd.header = new smevheader();
asd.header.MessageClass = "REQUEST";
asd.header.MessageID = "7D42D613-C09B-B42D-13BA-8FD259CEE61F";
asd.header.TimeStamp = "2012-05-25T09:44:32.663+04:00";
asd.Message.Date = DateTime.UtcNow.ToString("o");
asd.Message.Sender.Code = "ESIA01001";
asd.Message.Sender.Name = "МФЦ";
asd.Message.Recipient.Code = "ESIA01001";
asd.Message.Recipient.Name = "ТюмГУ";
asd.Message.TestMsg = "test msg";
asd.Message.ServiceName = "ConfirmationOfUTMNSStudentStatus";
asd.Message.TypeCode = "OTHR";
asd.Message.Status = "REQUEST";
asd.Message.ExchangeType = "0";
asd.MessageData.AppData.firstName = "Евгений";
asd.MessageData.AppData.middleName = "Олегович";
asd.MessageData.AppData.sureName = "Толпыго";
asd.MessageData.AppData.dateOfBirth = "02.06.1993";
asd.MessageData.AppData.documentCode = "1233 566234";
asd.MessageData.AppData.documentDate = "02.06.2007";
```

```
client.confirmStudentStatus(asd.header, ref asd.Message, ref asd.MessageData);
Console.WriteLine("Done");
Console.ReadKey();
```



Финансовая эффективность

- Стоимость разработки - 41 554,00р
- 60 заявок в месяц.
- Обработка одной заявки – 15 минут
- Стоимость часа работы замдиректора -89р
- Стоимость работ в месяц – 14 685,00р

Прибыль в течении двух лет -

- $352\ 440,00 - 45\ 154,00 = 307\ 286,00$ р
- Окупаемость проекта – 4 месяца

СПАСИБО ЗА ВНИМАНИЕ!!!!!!