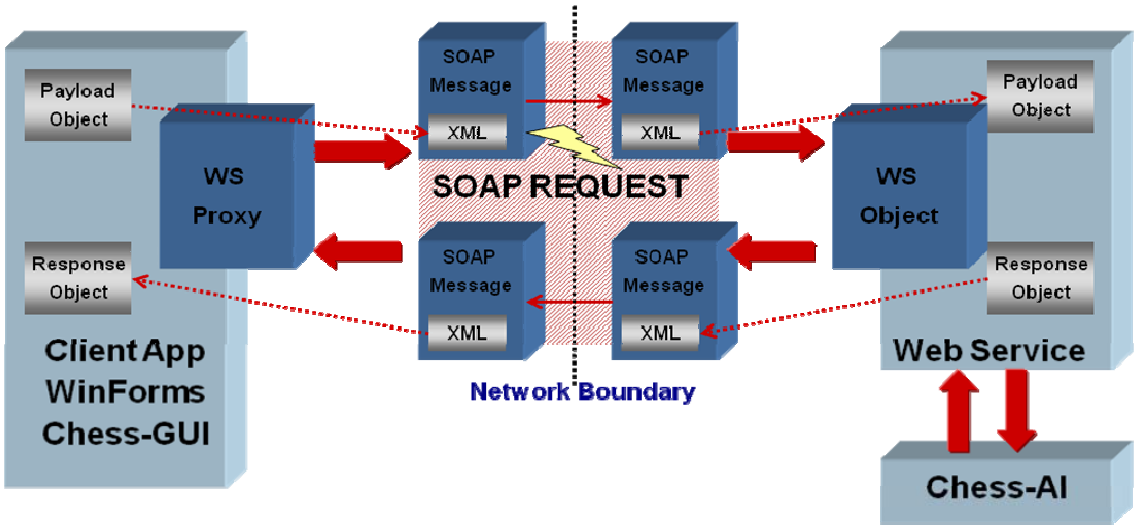
Assignment 3

For this last assignment you have to implement a XML Web Service with ASP.Net and C# and a WinForms client which access this Web Service. The Web Service should provide methods which return an appropriate move as a response to a given board configuration.
You can reuse the two main components from the last assignment: the AI component and your WinForms GUI. The Web Service is used as a proxy which calls the respective function from your AI component. On the other side you should reuse your WinForms GUI as a client which calls the XML Web Service after a human move to get a response from the AI on the server.
The following diagram shows the schematics of the workflow:



Use Visual Studio to create an ASP.Net Web Service Project. The generated application already contains a `HelloWorld()` method. To declare a method which is exposed via the Web Service you just have to add the `[WebMethod]` attribute before the method declaration. The following listing shows a simple example for a Web Method which receives a move as parameter and returns the response move:

```
using System;
using System.Data;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.ComponentModel;

namespace ChessService
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [ToolboxItem(false)]
```

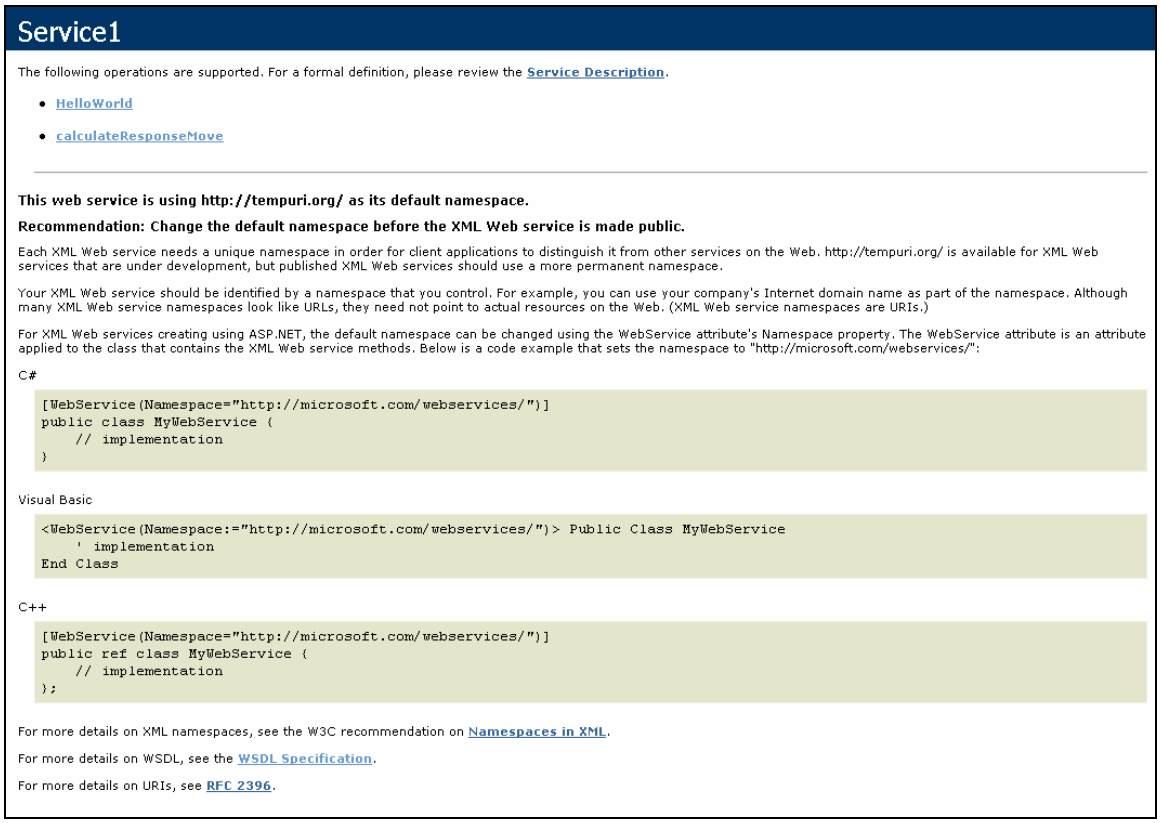
```
    public class Service1 : System.Web.Services.WebService
    {

        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }

        [WebMethod]
        public ChessMove calculateResponseMove(ChessMove move)
        {
            // ... calculate move ...
            ChessMove responseMove = AI.calcResponse(move);
            return responseMove;
        }
    }
}
```

Depending on the design of your AI component, you have to decide if you store the board on the client or on the web server. In the first case you always have to transmit the whole board and the Web Service returns the appropriate response move. In the later case the Web Service receives just a move and has to return the whole board. You are free to choose any architecture, which is appropriate for your system requirements. However, you have to provide a sketch of the workflow between your components and the rationale for your design in the accompanying report.

When you run the Web Service project, Visual Studio opens a web browser which points to the Web Service on the local host. The page could look as follows:



Service1

The following operations are supported. For a formal definition, please review the Service Description.

- HelloWorld
- calculateResponseMove

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML.

For more details on WSDL, see the WSDL Specification.

For more details on URIs, see RFC 2396.

To get the WSDL file click on the “Service Description” link in the first line. You also have to deliver this WSDL file of you final project. The WSDL description for the sample Web Service from is shown in the appendix.

We will code a sample Web Service and a WebForms client in the last exercise session.

Deadline:

August 1st

Deliverables and Points Assignment:

1. Report with rationale and workflow diagram [PDF] (6 points)
2. Web Service [ASP.Net Web Service Application and WSDL specification file] (7 points)
3. Client [WinForms Applicaiton] (7 points)

Appendix

WSDL description of the sample Web Service:

```
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
- <s:element name="HelloWorld">
  <s:complexType />
  </s:element>
- <s:element name="HelloWorldResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult" type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="calculateResponseMove">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="move" type="tns:ChessMove" />
  </s:sequence>
  </s:complexType>
  </s:element>
  <s:complexType name="ChessMove" />
- <s:element name="calculateResponseMoveResponse">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="calculateResponseMoveResult" type="tns:ChessMove" />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:schema>
  </wsdl:types>
- <wsdl:message name="HelloWorldSoapIn">
  <wsdl:part name="parameters" element="tns:HelloWorld" />
  </wsdl:message>
- <wsdl:message name="HelloWorldSoapOut">
  <wsdl:part name="parameters" element="tns:HelloWorldResponse" />
  </wsdl:message>
- <wsdl:message name="calculateResponseMoveSoapIn">
  <wsdl:part name="parameters" element="tns:calculateResponseMove" />
  </wsdl:message>
- <wsdl:message name="calculateResponseMoveSoapOut">
  <wsdl:part name="parameters" element="tns:calculateResponseMoveResponse" />
  </wsdl:message>
- <wsdl:portType name="Service1Soap">
- <wsdl:operation name="HelloWorld">
  <wsdl:input message="tns:HelloWorldSoapIn" />
  <wsdl:output message="tns:HelloWorldSoapOut" />
  </wsdl:operation>
- <wsdl:operation name="calculateResponseMove">
  <wsdl:input message="tns:calculateResponseMoveSoapIn" />
  <wsdl:output message="tns:calculateResponseMoveSoapOut" />
  </wsdl:operation>
  </wsdl:portType>
- <wsdl:binding name="Service1Soap" type="tns:Service1Soap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
  <soap:operation soapAction="http://tempuri.org/HelloWorld" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
  </wsdl:input>
- <wsdl:output>
  <soap:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
- <wsdl:operation name="calculateResponseMove">
  <soap:operation soapAction="http://tempuri.org/calculateResponseMove" style="document" />
- <wsdl:input>
  <soap:body use="literal" />
  </wsdl:input>
- <wsdl:output>
```

```
  <soap:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
- <wsdl:binding name="Service1Soap12" type="tns:Service1Soap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
  <soap12:operation soapAction="http://tempuri.org/HelloWorld" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
  </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
- <wsdl:operation name="calculateResponseMove">
  <soap12:operation soapAction="http://tempuri.org/calculateResponseMove" style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
  </wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
- <wsdl:service name="Service1">
- <wsdl:port name="Service1Soap" binding="tns:Service1Soap">
  <soap:address location="http://localhost:57168/Service1.asmx" />
  </wsdl:port>
- <wsdl:port name="Service1Soap12" binding="tns:Service1Soap12">
  <soap12:address location="http://localhost:57168/Service1.asmx" />
  </wsdl:port>
  </wsdl:service>
  </wsdl:definitions>
```